

Autocorrelation-Driven Diffusion Filtering

Michael Felsberg

Linköping University Post Print

N.B.: When citing this work, cite the original article.

©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Michael Felsberg , Autocorrelation-Driven Diffusion Filtering, 2011, IEEE Transactions on Image Processing, (20), 7, 1797-1806.

<http://dx.doi.org/10.1109/TIP.2011.2107330>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-65430>

Autocorrelation-Driven Diffusion Filtering

Michael Felsberg, *Member, IEEE*

Abstract—In this paper we present a novel scheme for anisotropic diffusion driven by the image autocorrelation function. We show the equivalence of this scheme to a special case of iterated adaptive filtering. By determining the diffusion tensor field from an autocorrelation estimate, we obtain an evolution equation that is computed from a scalar product of diffusion tensor and the image Hessian. We propose further a set of filters to approximate the Hessian on a minimized spatial support. On standard benchmarks, the resulting method performs favorable in many cases, in particular at low noise levels. In a GPU implementation video real-time performance is easily achieved.

Index Terms—Diffusion filtering, adaptive filtering, structure tensor, steerable filters, image enhancement
EDICS: TEC-PRC, TEC-PDE, TEC-RST

I. INTRODUCTION

THE efficient structure-preserving image denoising and enhancement is a relevant problem in many applications. In this paper, we propose a novel variant of tensor-driven diffusion filtering that: (a) only requires a scalar product of diffusion tensor and Hessian in the evolution equation, (b) is equivalent to a particular instance of adaptive filtering, and (c) is easy to implement in a fast numerical scheme.

Structure-preserving denoising and enhancement are practical problems to be solved in many applications, e.g. image and video coding, digital cameras, and medical imaging. Most of these applications require a real-time approach, which limits the range of available methods. The development of real-time capable approaches with state-of-the-art performance is still a relevant topic. The theoretical and practical insights formulated in this paper allow to achieve this goal, as it is shown in the experiments.

Structure-preserving denoising has a rather long history and non-linear diffusion [1] is probably one of the most commonly used techniques. The anisotropic extension of non-linear diffusion using the structure tensor [2] often leads to better results, in particular close to lines and edges. The numerical implementation of anisotropic diffusion is however less trivial as expected, as can be seen by the variety of algorithms and publications on the topic [3], [4], [5]. This is even more severe as we observed that the applied numerical scheme has larger influence on the quality of results than the choice of the method, i.e., using a sub-optimal numerical scheme for anisotropic diffusion results in worse peak-signal to noise ratios (PSNR) than a closer to optimal scheme for non-linear diffusion.

Besides non-linear diffusion, a large variety of approaches for image denoising have been published, too many to be covered here. Some of the more popular approaches use iterated adaptive filters [6], bilateral filtering [7], [8], mean-shift filtering [9], [10], channel decomposition [11], multi-band filtering [12], or block-based methods [13], [14]. Optimal results usually require estimates of image priors [15], [16], [17]. Some of the mentioned methods have been analyzed and compared in [15], [18], [19].

Besides the above cited work on diffusion, approaches to trace-based diffusion filtering have been suggested [20] and their relation to variational formulations and divergence expressions have been investigated [21]. The main difference of our approach to this previous work lies in the diffusion function and the spatial averaging in its computation. Tschumperlé and Deriche compute the diffusion tensor by averaging over different channels, i.e., in the case of grey scale images, no averaging is done at all. In that case, the diffusion tensor is at most of rank 1 and the processing across and parallel to the image gradient are not independent. The processing becomes independent if the image Hessian is used to compute the diffusivity, even if it is not spatially averaged [20]. Computing the diffusivity from the structure tensor *with spatial averaging*, however, results in a third alternative to trace-based diffusion [22], which has a rank 2 diffusivity tensor *and* is based on first order derivatives.

In the present work, the latter approach is modified by considering the diffusion tensor as a function of the autocorrelation function. In particular, we use the Hessian, which represents the curvature of the autocorrelation function, to control the diffusion process. This is intuitively motivated by the fact that the curvature (flat, parabolic, or elliptic) represents the type of local structure (flat, linear, or two-dimensional) to a large extend independently of intensity or texture. Flat structures should be diffused isotropically, linear structures only in one direction, and two-dimensional structures not at all. Furthermore, we derive new discrete filter masks, apply a new noise estimation method [23], and extend the experimental evaluation using methods that take into account the visual quality: the visual information fidelity (VIF) [24] and the structural similarity index (SSIM) [25].

The paper is structured as follows. In Sect. II, we introduce some tensor notation and calculus and we give some background on the structure tensor, anisotropic diffusion, and adaptive filtering. In Sect. III, we derive the novel scheme from the anisotropic diffusion equation and relate it to iterated adaptive filtering, we propose concrete discrete filters for all relevant steps and discuss practical aspects of the algorithm. In Sect. IV, we present a number of standard denoising experiments in order to compare our approach with methods from the literature. Finally, we discuss the advantages and

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

M. Felsberg is with the Computer Vision Laboratory, Department of Electrical Engineering, Linköping University, S-58183 Linköping, Sweden, michael.felsberg@liu.se.

drawbacks of the proposed method in the concluding Sect. V.

II. TENSORS, DIFFUSION, AND ADAPTIVE FILTERING

In this section we introduce some notation, terms, and methods from the literature, which will be required in subsequent sections.

A. Notation and Tensor Calculus

Despite the fact that tensors are coordinate-system independent algebraic entities, we restrict ourselves to matrix representations of tensors in this paper. In what follows, (real) vectors are denoted as bold letters and matrices as bold capital letters. The scalar product between two column vectors \mathbf{a} and \mathbf{b} is usually denoted by the matrix product of the transpose of \mathbf{a} and \mathbf{b} : $\mathbf{a}^T \mathbf{b}$. Tensors (of second order) will be often built by outer products of vectors, e.g., $\mathbf{T} = \mathbf{a} \mathbf{b}^T$. Since second order tensors form a vector space, a scalar product (Frobenius product) is defined, which reads for two tensors \mathbf{A} and \mathbf{B}

$$\langle \mathbf{A} | \mathbf{B} \rangle = \text{trace}(\mathbf{A} \mathbf{B}^T) = \sum_i \sum_j a_{ij} b_{ij} . \quad (1)$$

In most cases we will consider symmetric tensors, i.e., $a_{ij} = a_{ji}$ and for these tensors the spectral theorem allows a decomposition into real eigenvalues:

$$\mathbf{A} = \mathbf{O} \mathbf{\Lambda} \mathbf{O}^T = \mathbf{O} \text{diag}(\lambda_i) \mathbf{O}^T \quad (2)$$

where \mathbf{O} is an orthogonal matrix formed by the eigenvectors of \mathbf{A} and $\mathbf{\Lambda}$ is the diagonal matrix containing the corresponding (real) eigenvalues λ_i . Note that arbitrary powers of \mathbf{A} can be computed as

$$\mathbf{A}^n = (\mathbf{O} \mathbf{\Lambda} \mathbf{O}^T)^n = \mathbf{O} \mathbf{\Lambda}^n \mathbf{O}^T = \mathbf{O} \text{diag}(\lambda_i^n) \mathbf{O}^T \quad (3)$$

and due to linearity

$$\exp(k \mathbf{A}) = \mathbf{O} \text{diag}(\exp(k \lambda_i)) \mathbf{O}^T . \quad (4)$$

In many cases we deal with vector-valued or tensor-valued functions in this paper, even if written as vectors or tensors. The spatial variable, mostly denoted as \mathbf{x} , is often omitted. In this context, differential operators are based on the partial derivatives with respect to the components of \mathbf{x} and are denoted using the nabla operator ∇ , such that

$$\nabla b = \left(\frac{\partial b}{\partial x_i} \right)_i \quad (5)$$

$$\nabla^T \mathbf{a} = \text{div}(\mathbf{a}) = \sum_i \frac{\partial a_i}{\partial x_i} \quad (6)$$

$$\nabla^T \mathbf{A} = \text{div}(\mathbf{A}) = \left(\sum_i \frac{\partial a_{ij}}{\partial x_i} \right)_j . \quad (7)$$

B. Structure Tensor

Tensor representations occur frequently in image processing, mostly for representing local orientation and the deviation from a simple signal model [6]. The gradient based structure tensor can be derived as the solution of a least-squares approach to fitting a simple signal model [26]. Another alternative is to consider the structure tensor as an estimate of

the Hessian of the autocorrelation function of the signal [27]. Assuming ergodicity, the autocorrelation function of a 2D image $b(\mathbf{x})$ is given by correlating the image with itself

$$R = b \star b - b_0^2 \quad (8)$$

where b_0 denotes the DC part of b . Since R is a symmetric function, we might expand it as

$$R(\mathbf{x}) = R(0) + \frac{1}{2} \mathbf{x}^T \mathbf{H} R \mathbf{x} + \mathcal{O}(|\mathbf{x}|^4) \quad (9)$$

where \mathbf{H} denotes the Hessian operator. With the power theorem we obtain (see [28], p. 317)

$$\mathbf{H} R = E[\nabla b \nabla^T b] \quad (10)$$

where $E[\cdot]$ denotes the expectation value. This expectation value is usually estimated by locally weighted averaging the outer product of the image gradient [29], [30]:

$$\mathbf{J}(\mathbf{x}) = \int w(\mathbf{y}) \nabla b(\mathbf{x} - \mathbf{y}) \nabla^T b(\mathbf{x} - \mathbf{y}) d\mathbf{y} . \quad (11)$$

Note that the gradient operators in the equation above are normally regularized by some low-pass filter, e.g., using Gaussian derivatives. The local weight function w is also mostly chosen as a Gaussian function.

C. Anisotropic Diffusion

We will not give a detailed introduction to anisotropic diffusion, the reader is referred to, e.g., the work of Weickert [31]. The diffusion scheme is defined as an evolution equation of an image $b(\mathbf{x})$ over time t as:

$$\frac{\partial b}{\partial t} = \text{div}(\mathbf{D}(\mathbf{J}) \nabla b) \quad (12)$$

where $\mathbf{D}(\mathbf{J})$ is the diffusion tensor, which is computed by modifying the eigenvalues of the structure tensor \mathbf{J} . Applying the product rule, the divergence term is split into two parts (cf. (7))

$$\text{div}(\mathbf{D}(\mathbf{J}) \nabla b) = \text{div}(\mathbf{D}(\mathbf{J})) \nabla b + \text{trace}(\mathbf{D}(\mathbf{J}) \mathbf{H} b) . \quad (13)$$

If the structure tensor is computed by spatial averaging (11), the divergence of the diffusion tensor is non-zero in general and needs to be estimated. If the tensor is computed by averaging over the different dimensions of a vector-valued image, i.e., no spatial averaging is applied, the divergence term can be converted into a trace-based term [21], [20]. Thus, by modifying $\mathbf{D}(\mathbf{J})$, the evolution equation can be expressed by a single trace-based term, i.e., the divergence of the diffusion tensor is eliminated. However, according to our knowledge, this elimination is impossible if the structure tensor is computed by spatial averaging.

When implementing anisotropic diffusion, an iterative algorithm that updates the input image successively has to be implemented. In each time-step, the diffusion tensor and the image gradient have to be estimated. If the evolution is divergence-based, another numerical approximation of a derivative operator is required, which is applied to the product of diffusion tensor and gradient. Hence, five sequential operations are required: image gradients, point-wise products, local

averaging, point-wise product, and gradient. If the evolution is reformulated as a trace-based approach, only four sequential operations are required, since the final gradient is replaced with a second image derivative which can be computed in parallel to the diffusion tensor.

D. Adaptive Filtering

Adaptive filtering is a more general and actually earlier published variant of steerable filters [32], developed by Knuttson et al. [33], [34]. The main idea is to compose a spatially variant filter kernel by linear combinations of shift invariant kernels. The linear coefficients are locally estimated by an orientation-dependent scheme. A comprehensible formulation of the method can be found in [6], Chapt. 10. Adopting the notation above, the filter is composed as

$$h_{\text{adapt}} = h_{\text{LP}} + \sum_k \langle \mathbf{C}(\mathbf{J}) | \tilde{\mathbf{N}}_k \rangle h_{\text{HP},k} \quad (14)$$

where $h_{\text{HP},k}$ is an orientation selective high-pass filter with orientation \mathbf{n}_k , $\tilde{\mathbf{N}}_k$ is the dual tensor for the orientation tensor $\mathbf{n}_k \mathbf{n}_k^T$ and \mathbf{C} is the control tensor. The high-pass filter with orientation \mathbf{n}_k is defined in the Fourier domain by a polar separable filter with radial component $\rho(|\mathbf{u}|)$ and an angular component $D(\mathbf{u}/|\mathbf{u}|) = (\mathbf{n}_k^T \mathbf{u})^2 / |\mathbf{u}|^2$. Consider for instance the case $\rho(|\mathbf{u}|) = |\mathbf{u}|^2$ such that

$$H_{\text{HP},k} = (\mathbf{n}_k^T \mathbf{u})^2 \quad (15)$$

with $\mathbf{n}_1 = (1, 0)^T$, $\mathbf{n}_2 = (0, 1)^T$, and $\mathbf{n}_3 = (\sqrt{1/2}, \sqrt{1/2})^T$. The resulting frequency responses are $H_{\text{HP},1} = u_1^2$, $H_{\text{HP},2} = u_2^2$, and $H_{\text{HP},3} = u_1^2/2 + u_1 u_2 + u_2^2/2$.

The corresponding orientation tensors are given as

$$\mathbf{N}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{N}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathbf{N}_3 = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$$

The dual tensors can be computed via the isometric vector representation as

$$\tilde{\mathbf{N}}_1 = \begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix} \quad \tilde{\mathbf{N}}_2 = \begin{bmatrix} 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix} \quad \tilde{\mathbf{N}}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} .$$

Finally, in our example, the rightmost term in (14) is proportional to the negative Hessian since

$$\sum_k \tilde{\mathbf{N}}_k H_{\text{HP},k} = \mathbf{u} \mathbf{u}^T . \quad (16)$$

The tensor controlled adaptive filter scheme is often applied in an iterated way to achieve good denoising results [6], Chapt. 10. By selecting the high-pass filter in the way we have done in the example above, but with a very small constant multiplier $0 < \gamma \ll 1$ and selecting the low-pass filter appropriately as

$$H_{\text{LP}} = 1 - \gamma \rho(|\mathbf{u}|) = 1 - \gamma |\mathbf{u}|^2 , \quad (17)$$

it is possible to show that the iterated adaptive filtering implements a numerical scheme for autocorrelation-driven anisotropic diffusion, see Sect. III-B.

III. AUTOCORRELATION-DRIVEN DIFFUSION FILTERING

This section starts with defining autocorrelation-driven diffusion filtering in the continuous domain, where the autocorrelation function estimate is inserted after computing the divergence, resulting in a purely trace-based formulation. This continuous equation is then discretized in the temporal domain and an equivalence to iterated adaptive filtering is found. Finally, the scheme is fully discretized, its stability is analyzed, and practical aspects are discussed.

A. Continuous Autocorrelation-Driven Diffusion Filtering

Many schemes for tensor-driven anisotropic diffusion use the structure tensor for controlling the diffusion process. Interpreting the structure tensor as an estimate of the autocorrelation function, cf. Sect. II-B (9) to (11), one can go one step back and reformulate anisotropic diffusion with a diffusion tensor that depends on the autocorrelation function instead:

Definition 1: The autocorrelation-driven anisotropic diffusion is defined by the evolution

$$\frac{\partial b}{\partial t} = \text{div}(\mathbf{B}(R) \nabla b) , \quad (18)$$

where $\mathbf{B}(R)$ is the diffusion tensor determined by the autocorrelation function R at the current spatial position.

This definition is particularly sensible from a statistical point of view if we consider the ergodicity assumption which is essential for denoising by weighted spatial averaging. If the image signal is ergodic in a strict sense, all statistical moments are stationary and thus the autocorrelation is constant, resulting in unweighted averaging. If the image signal is structured, its statistical moments are not stationary, and thus, it is not ergodic and should not be averaged spatially. The autocorrelation function becomes highly peaked in all directions in this case. If the image signal is ergodic in one direction, but structured in the orthogonal direction (e.g., edge or line structure), averaging should only be performed orthogonal to the structure. The autocorrelation function is peaked along the structured direction and flat orthogonal to it.

In this work, we consider anisotropic diffusion (12) as an approximation to (18). We will now derive another approximation to (18) which is of fourth order and which always guarantees an even autocorrelation function:

Theorem 1: Assume that \mathbf{J} according to (11) is at least a second order approximation to $\mathbf{H}R$ and \mathbf{B} is continuously differentiable. Then

$$\frac{\partial b}{\partial t} = \langle \mathbf{D}(\mathbf{J}) | \mathbf{H}b \rangle \quad (19)$$

is a fourth order approximation to (18) for a suitable function \mathbf{D} .

Proof: For the diffusivity $\mathbf{B}(R)$, a fourth order approximation is given as

$$\mathbf{B}(R) \approx \mathbf{B}(R(0)) + \frac{1}{2} \mathbf{x}^T \mathbf{H}R \mathbf{x} \mathbf{B}'(R(0)) . \quad (20)$$

Since \mathbf{J} is at least a second order approximation of $\mathbf{H}R$,

$$\mathbf{D}(\mathbf{J}) \doteq \mathbf{B}(R(0)) + \frac{1}{2} \mathbf{x}^T \mathbf{J} \mathbf{x} \mathbf{B}'(R(0)) \quad (21)$$

is also a fourth order approximation of $\mathbf{B}(R)$. If we now apply the product rule to (18), we obtain with (1)

$$\operatorname{div}(\mathbf{B}(R)\nabla b) = \operatorname{div}(\mathbf{B}(R))\nabla b + \langle \mathbf{B}(R)|\mathbf{H}b \rangle . \quad (22)$$

For the first term, we obtain

$$\operatorname{div}(\mathbf{B}(R)) = \mathbf{B}'\nabla R \quad (23)$$

which vanishes in the origin since R is an even function. Hence,

$$\operatorname{div}(\mathbf{B}(R)\nabla b) = \langle \mathbf{B}(R)|\mathbf{H}b \rangle , \quad (24)$$

and plugging in (21) we obtain that (19) is a fourth order approximation to (18). ■

If we plug the autocorrelation approximation (21) into (18) before computing the divergence, we obtain the standard formulation of anisotropic diffusion (12). The only difference to (19) is that the divergence of the diffusion tensor does not vanish in general:

$$\operatorname{div}(\mathbf{D}(\mathbf{J})) \neq 0 .$$

However, this implies that the autocorrelation function contains a first order term $\frac{\partial R}{\partial x}$, leading to a third order approximation to (18). Note that this result is just on the approximation order and assumes that (18) is our model instead of (12).

B. A Time-Discrete Scheme for Autocorrelation-Driven Anisotropic Diffusion

In order to obtain a filtering method, the continuous differential equation (19) needs to be discretized in the spatial and temporal domain. We start with discretizing the temporal domain, since we aim at relating diffusion filtering to adaptive filtering, which has been introduced with continuous notation in Sect. II-D. Using a standard explicit scheme for time discretization, we obtain from (19)

$$b_{t+1} = b_t + \gamma \langle \mathbf{D}(\mathbf{J})|\mathbf{H}b_t \rangle \quad (25)$$

where we use $\gamma = 0.05$ throughout this paper. The upper limit for γ depends on the spatial discretization and is discussed in Sect. III-D.

For the iterative filtering (25), we obtain

Theorem 2: Autocorrelation-driven diffusion filtering according to (25) and the particular case of iterated adaptive filtering using (15) are equivalent.

Proof: As it has been shown in the example in Sect. II-D, the Hessian can be written in terms of a set of dual tensors and corresponding filters (16). Hence, $\sum_k \tilde{\mathbf{N}}_k h_{\text{HP},k} = -\mathbf{H}$ and plugging this into (25) gives

$$b_{t+1} = b_t + \gamma \left\langle \mathbf{Id} - \mathbf{C}(\mathbf{J}) \middle| - \sum_k \tilde{\mathbf{N}}_k h_{\text{HP},k} b_t \right\rangle \quad (26)$$

where we replaced $\mathbf{D}(\mathbf{J}) = \mathbf{Id} - \mathbf{C}(\mathbf{J})$. Sorting terms gives

$$b_{t+1} = \left(\delta - \gamma(h_{\text{HP},1} + h_{\text{HP},2}) + \sum_k \langle \mathbf{C}(\mathbf{J})|\tilde{\mathbf{N}}_k \rangle \gamma h_{\text{HP},k} \right) b_t \quad (27)$$

which is equivalent to an adaptive filtering according to (14) with constant multiplier γ and the low-pass filter $\delta - \gamma(h_{\text{HP},1} + h_{\text{HP},2})$. This low-pass filter is the inverse Fourier transform of (17). ■

C. A Fully Discrete Scheme for Autocorrelation-Driven Anisotropic Diffusion

In order to derive a fully discrete scheme for (25), we also discretize the spatial domain. Three filter (sets) have to be designed: The Hessian, the derivatives in (11), and the averaging kernel in (11). The latter is chosen to be a Gaussian function with variance $\sigma^2 = 1$ throughout the paper. The exact value for the variance is not crucial, as the experimental results do not differ for small variations. The derivatives used in the structure tensor (11) and the second derivatives used to compute the Hessian are computed with filter masks of minimum size. This is motivated by an observation made in [35]: Reducing linear regularization, i.e., using smaller filter masks, for the benefit of increased regularization of the structure tensor improves the overall result. In order to obtain small filter masks with good isotropy, we apply a similar optimization strategy as in [36], but with a different objective function. We start by assuming 3×3 filter masks, which gives us one degree of freedom for the first derivatives ($p \in [0; 0.25]$ since we want to avoid extra zeros in the frequency response):

$$\frac{\partial b}{\partial x_1} \approx \frac{1}{2}([p \ 1 - 2p \ p]^T[-1 \ 0 \ 1]) * b \quad (28)$$

$$\frac{\partial b}{\partial x_2} \approx \frac{1}{2}([-1 \ 0 \ 1]^T[p \ 1 - 2p \ p]) * b . \quad (29)$$

A second order derivative approximation can also be computed with a 3×3 filter, which we derive from one-sided finite differences. This guarantees frequency responses with zero DC level and without extra zeros. Starting with the second derivative for the horizontal coordinate, the 1D Laplace filter is obtained by combining a left-sided and a right-sided difference:

$$\frac{\partial^2}{\partial x_1^2} \approx [-1 \ 1 \ 0] * [0 \ -1 \ 1] = [1 \ -2 \ 1] . \quad (30)$$

A second approximation without extra zeros is obtained by the 2D filters that average in the vertical direction with a 2-box filter (see also [5]):

$$\begin{aligned} \frac{\partial^2}{\partial x_1^2} &\approx \frac{1}{4} \begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & -1 & 1 \end{bmatrix} \\ &= \frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{bmatrix} . \end{aligned} \quad (31)$$

The actually used filters are the weighted average of both approximations (30) and (31), which span the whole space of zero DC 3×3 second derivatives without extra zeros in the frequency domain:

$$\frac{\partial^2}{\partial x_1^2} \approx \frac{1}{4} \begin{bmatrix} q & -2q & q \\ 4 - 2q & -8 + 4q & 4 - 2q \\ q & -2q & q \end{bmatrix} \quad (32)$$

$$\frac{\partial^2}{\partial x_2^2} \approx \frac{1}{4} \begin{bmatrix} q & 4 - 2q & q \\ -2q & -8 + 4q & -2q \\ q & 4 - 2q & q \end{bmatrix} \quad (33)$$

The mixed derivatives corresponding to (30) and (31) are identical:

$$\frac{\partial^2}{\partial x_1 \partial x_2} \approx \frac{1}{4} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}^T = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}. \quad (34)$$

The optimal filterset in terms of p and q is obtained by minimizing the weighted angular error between two normalized tensors in the Fourier domain:

$$\mathbf{T}_1 = \frac{1}{H_1^2 + H_2^2} \begin{bmatrix} H_1^2 & H_1 H_2 \\ H_1 H_2 & H_2^2 \end{bmatrix} \quad (35)$$

$$\mathbf{T}_2 = -\frac{1}{\sqrt{H_{11}^2 + H_{22}^2 + 2H_{12}^2}} \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix}, \quad (36)$$

where \mathbf{u} is the 2D frequency, $H_k = H_k(\mathbf{u}; p)$ are the Fourier transforms of $\frac{\partial}{\partial x_k}$ and $H_{kl} = H_{kl}(\mathbf{u}; q)$ are the Fourier transforms of $\frac{\partial^2}{\partial x_k \partial x_l}$. The weighting function $v = v(|\mathbf{u}|)$ is determined by the weight w in (11) and the statistics of natural images ($|\mathbf{u}|^{-1}$). Hence, we minimize

$$\min_{p,q} \int v \cos^{-1} (\langle \mathbf{T}_1 | \mathbf{T}_2 \rangle)^2 d\mathbf{u}. \quad (37)$$

The resulting parameters p and q are given in good approximation as $p = \frac{1}{4}$, i.e., a Sobel filter for the first derivative, and $q = \frac{13}{16}$.

D. Stability of the Discrete Scheme

Conditions for the stability of discrete diffusion schemes have been proposed in [37]. The fully discrete scheme is rewritten in the form

$$b_{t+1} = Q(b_t) b_t \quad (38)$$

and the function $Q(b_t)$ is tested for six criteria (D1)–(D6): continuity in its argument, symmetry, unit row sums, nonnegative off-diagonals, irreducibility, and positive diagonal. In our case, we derive a parametrized effective convolution kernel, which can then be converted into the corresponding matrix operator $Q(b_t)$.¹

First of all, we need to specify the diffusivity function $\mathbf{D}(\mathbf{J})$. It is known from the literature that the diffusivity determines the relation between non-linear diffusion and robust estimation [17], [38]. The width of the influence function in robust estimation is more important than the exact shape [38], such that we decided to choose an algebraically simple diffusivity of exponential form:

$$\mathbf{D}(\mathbf{J}) = \exp\left(-\frac{1}{k} \mathbf{J}\right). \quad (39)$$

Due to (4), this corresponds to applying the corresponding exponential mapping to the eigenvalues. Without loss of generality, we assume $k = 1$ for the remainder of this section.

¹This actually means that we assume an infinitely large or periodic discrete image. For finite domains, $Q(b_t)$ is no longer equivalent to a convolution kernel.

Let $\lambda_1 > \lambda_2$ denote the eigenvalues of \mathbf{J} and θ the angle of the eigenvector corresponding to λ_1 . Hence, using $\sigma_k = \exp(-\lambda_k)$ (note $\sigma_2 > \sigma_1$)

$$\begin{aligned} \mathbf{D}(\mathbf{J}) &= \begin{bmatrix} d_{11} & d_{12} \\ d_{12} & d_{22} \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1 \cos^2 \theta + \sigma_2 \sin^2 \theta & -(\sigma_2 - \sigma_1) \cos \theta \sin \theta \\ -(\sigma_2 - \sigma_1) \cos \theta \sin \theta & \sigma_1 \sin^2 \theta + \sigma_2 \cos^2 \theta \end{bmatrix} \end{aligned} \quad (40)$$

and we obtain (using $\alpha = \sigma_1 + \sigma_2 = d_{11} + d_{22}$)

$$\langle \mathbf{D}(\mathbf{J}) | \mathbf{H} \rangle = \begin{bmatrix} \alpha q + 2d_{12} & 4d_{22} - 2\alpha q & \alpha q - 2d_{12} \\ 4d_{11} - 2\alpha q & 4\alpha(q - 2) & 4d_{11} - 2\alpha q \\ \alpha q - 2d_{12} & 4d_{22} - 2\alpha q & \alpha q + 2d_{12} \end{bmatrix} \quad (41)$$

Finally, the filter that corresponds to $Q(b_t)$ is given as $\delta(\mathbf{x}) + \gamma \langle \mathbf{D}(\mathbf{J}) | \mathbf{H} \rangle$, cf. (25).

For $Q(b_t)$, we immediately get continuity, symmetry, unit row sums, and irreducibility. Positive diagonal entries (D6) are obtained if

$$\gamma < \frac{2}{19}. \quad (42)$$

The off-diagonals are however not nonnegative (D4), choose e.g. $\sigma_2 = 1$, $\sigma_1 = 1/15$ and $\theta = \pi/4$, the upper left element in (41) equals $-1/15$. Even though the nonnegativity is violated, this does not imply instability of the scheme.

In particular, for $q = 0$, the proposed scheme is identical to the explicit scheme [39], which is known to be conditional stable [40]. If we restrict the stability analysis to locally simple signals, we can draw conclusions concerning stability. For this class of signals, the structure tensor \mathbf{J} is of rank one. Hence, $\sigma_1 \ll 1$ and $\sigma_2 = 1$. Assuming $\sigma_1 = 0$, we can compute the θ -intervals where the coefficients in (41) become negative. These are (modulo π):

$$(-0.22\pi; 0.22\pi) \quad \text{for} \quad 4d_{22} - 2\alpha q \quad (43)$$

$$(0.15\pi; 0.35\pi) \quad \text{for} \quad \alpha q + 2d_{12} \quad (44)$$

$$(0.28\pi; 0.72\pi) \quad \text{for} \quad 4d_{11} - 2\alpha q \quad (45)$$

$$(0.65\pi; 0.85\pi) \quad \text{for} \quad \alpha q - 2d_{12} \quad (46)$$

These intervals are such that the angular distance to the gradient orientation is maximized, i.e., we have a maximum level of integration in these sectors and due to the vanishing sums property, the resulting effective operator is nonnegative. For instance, if we integrate along the columns (assuming $\theta = 0$) of (41), we obtain the weighted 1D Laplacian operator $\sigma_1[4, -8, 4]$. It is also worthwhile noticing that for the case $q = 0$, the first (43) and the third sector (45) vanish, but the second (44) and fourth (46) cover the full respective quadrant. For $q = 1$, the first (43) and third sector (45) cover the full respective quadrant and the second (44) and fourth sector (46) vanish.

E. Practical Details

What we experienced as the most critical parameter of the algorithm in order to obtain comparative results, is the width of the diffusivity function (39) k . Too large values of k lead to blurred results, i.e., structure is considered as noise, and too small values lead to noisy results, since noise is considered

as structure. The parameter that gives the statistically best discrimination of structure and noise can be derived from a null-hypothesis test and the resulting choice for k is [41]

$$k = \frac{e-1}{e-2} \sigma_e^2 \quad (47)$$

where σ_e^2 is the *estimated noise variance* according to the method described in [23] with a minor modification. The estimation is based on a mixture of two χ_2^2 distributions, which are fitted to the squared gradient magnitude distribution using the EM algorithm. The minor modification that we have applied here is, that we swap the two mixture components during the iterations if the noise has a larger estimated variance than the signal. Note further that the variance is given as twice the mixture parameter μ of the χ_2^2 distribution: $\sigma_e^2 = 2\mu$.

When implementing the diffusivity function (39), it appears to be preferable to apply the exponential mapping directly on the matrix instead of an eigenvalue manipulation. However, the latter is faster since the eigenvalues can be extracted in closed form [5], [26]. In principle this would also allow to combine explicitly isotropic and anisotropic diffusion [42], but we chose to apply the same scheme to edge-like images (e.g. the pepper image) and fingerprint images since the achieved results were satisfying in both cases. Our scheme shares similarities with edge-enhancing diffusion, which explains good performance on edge-like images. Good performance on line-like structures might be explained by the missing odd order terms in the scheme due to omitting the divergence of the diffusion tensor.

Finally, an often discussed parameter of anisotropic diffusion is the *stopping time*. We will not give a detailed consideration here. In our implementation the iterations are stopped as soon as the image update is below a certain threshold. Based on k , the threshold for the stopping criterion is empirically set to $k^{1/5}/40$ by looking at curves of maximum PSNR in logarithmic scale. This is however only partly satisfying as the rate of change might depend on the image structure and not on the difference to a noise-free instance of the image. The stopping time should rather be based on the estimated noise level at each time instance. In practice, however, we observe little difference of the two strategies, since the largest area of the images is rather flat and the mean or median image update is basically given by the amount of removed noise (which is in itself directly related to the amount of noise).

IV. EXPERIMENTS

This section consists of two parts: the experiments and the discussion of results.

A. Experimental Results

For the experiments, we used a popular set of images that has already been used in earlier publications [12], [43], [44]. The evaluation criteria are, besides the PSNR, SSIM [25] and VIF [24]. The PSNR does not reflect what kind of artifacts are introduced, cf. Fig. 1, therefore, the other two criteria have been included.

Two of the most successful methods in terms of PSNR are due to Portilla et al. (GSM) [12] and Dabov et al.

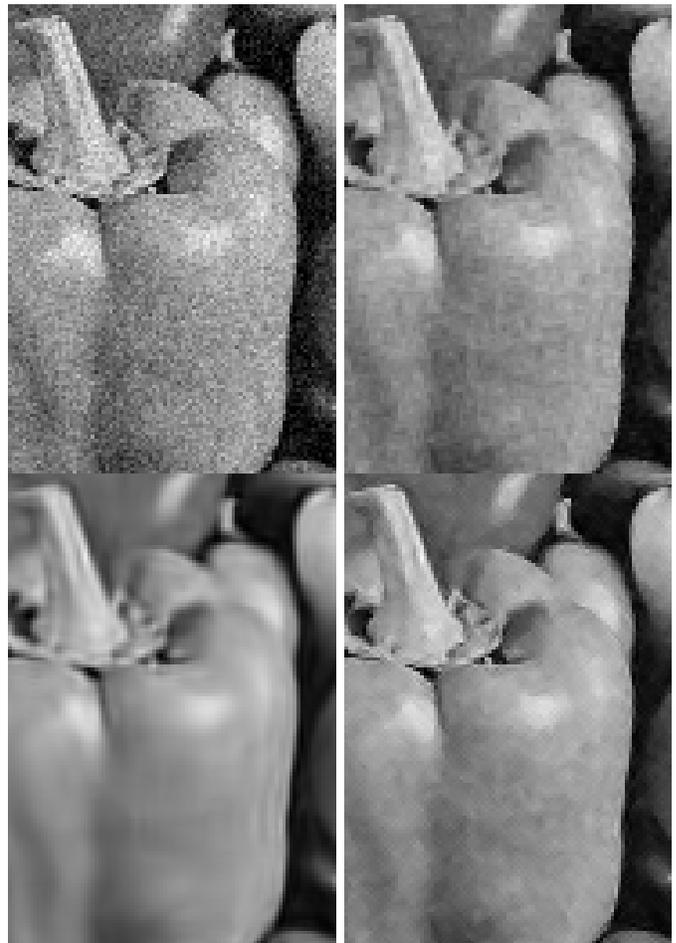


Fig. 1. The PSNR does not reflect the type of residual error. Here: detail of the pepper image ($\sigma = 15$, top left) for which the proposed method, GSM, and EEDF yield about the same PSNR result. Top right: EEDF, bottom left: GSM, and bottom right: proposed method.

(BM3D) [14], so that we compare our method with their implementations. Other methods as FoE [43] and K-SVD [13] are respectively slightly inferior [15].² Furthermore, we compare our results to a standard implementation of anisotropic (edge enhancing) diffusion filtering (EEDF) [3]. We evaluate each image and noise level with ten different instances of noise, clipped and quantized to eight bit integers. Each method is fed with an estimated noise level and not with the ground truth noise level of the added noise as done in the original evaluations [12], [15]. The noise level is used as a parameter in all four methods to determine the degree of smoothing. Hence, results for GSM and BM3D differ from the literature, since the original images already contain noise, resulting in a higher noise level estimate.

For the smoothed result, PSNR, SSIM, and VIF values are reported. Our method is entirely implemented in Matlab and is about 6 times faster than GSM, which is a combination of Matlab and C code. Our implementation is similar to the EEDF implementation [3] and shares basically the same parameters. Therefore, we use the same parameter settings for the EEDF,

²The method in [15] seems to work slightly better than [14], but the code for the latter was more straightforward to be used for denoising.

cf. Section III-E. All our attempts to improve the results from EEDF by varying the parameters have not lead to a systematic increase of the quality.

The resulting PSNRs for the four methods are plotted in Fig. 2 for two different images. Numerical results of the

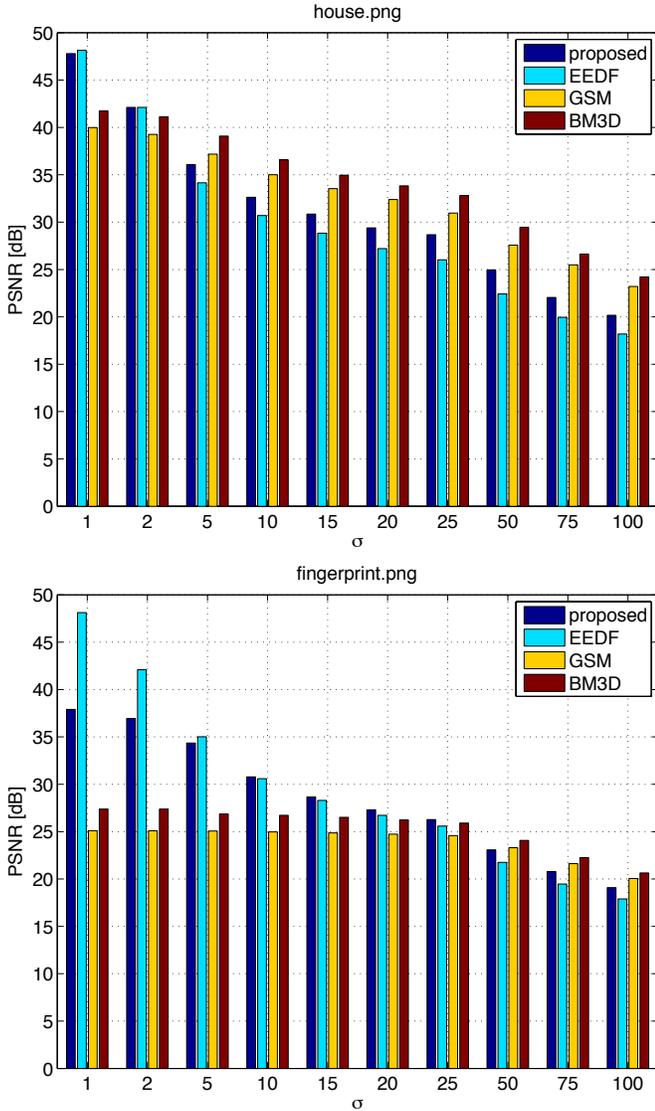


Fig. 2. PSNR results for the proposed method, EEDF, GSM, and BM3D for the house image and the fingerprint image.

proposed method are summarized in Table I.

The resulting VIFs and SSIMs for the four methods are plotted in Fig. 3 and Fig. 4, respectively. Numerical results of the proposed method are summarized in Table II for the VIF and in Table III for the SSIM. In order to allow a more systematic analysis of the performance, the differences between the proposed method and the three competing methods are plotted in Fig. 5.

B. Discussion of Results

As it can be seen from the PSNRs in Fig. 2 and the solid line plots in Fig. 5 (top) for low noise levels, the proposed

TABLE I
PSNRs FOR THE PROPOSED METHOD

σ	Lena	Barb.	Boat	House	Pepper	Fingerp.	Flinst.
1	47.8	47.8	46.6	47.8	48.1	37.9	38.5
2	42.5	42.8	42.5	42.1	42.8	36.9	37.4
5	37.5	36.6	36.1	36.1	36.9	34.4	34.8
10	33.4	32.3	32.3	32.6	33.2	30.8	31.3
15	31.6	29.1	30.9	30.9	30.7	28.7	29.9
20	30.4	26.5	29.6	29.4	29.6	27.3	28.4
25	29.7	24.9	28.5	28.7	28.9	26.3	27.0
50	25.8	22.9	24.9	25.0	24.1	23.1	21.2
75	22.8	21.1	22.1	22.0	21.4	20.8	19.5
100	20.5	19.3	20.0	20.2	19.4	19.1	17.7

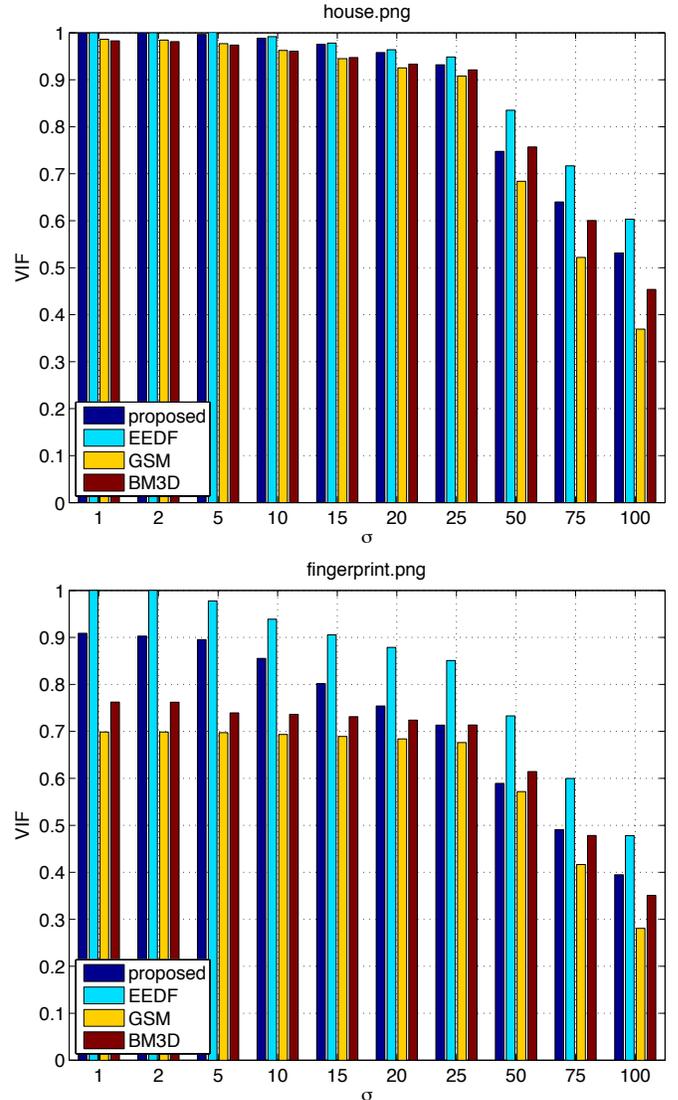


Fig. 3. VIF results for the proposed method, EEDF, GSM, and BM3D for the house image and the fingerprint image.

method works better than GSM and BM3D and slightly worse than EEDF. For high noise levels, GSM and BM3D work slightly better than the proposed method and EEDF works worst. Integrated over the different noise levels, GSM works only better than the proposed method for the house image, BM3D works worse for boat, pepper, and fingerprint, and

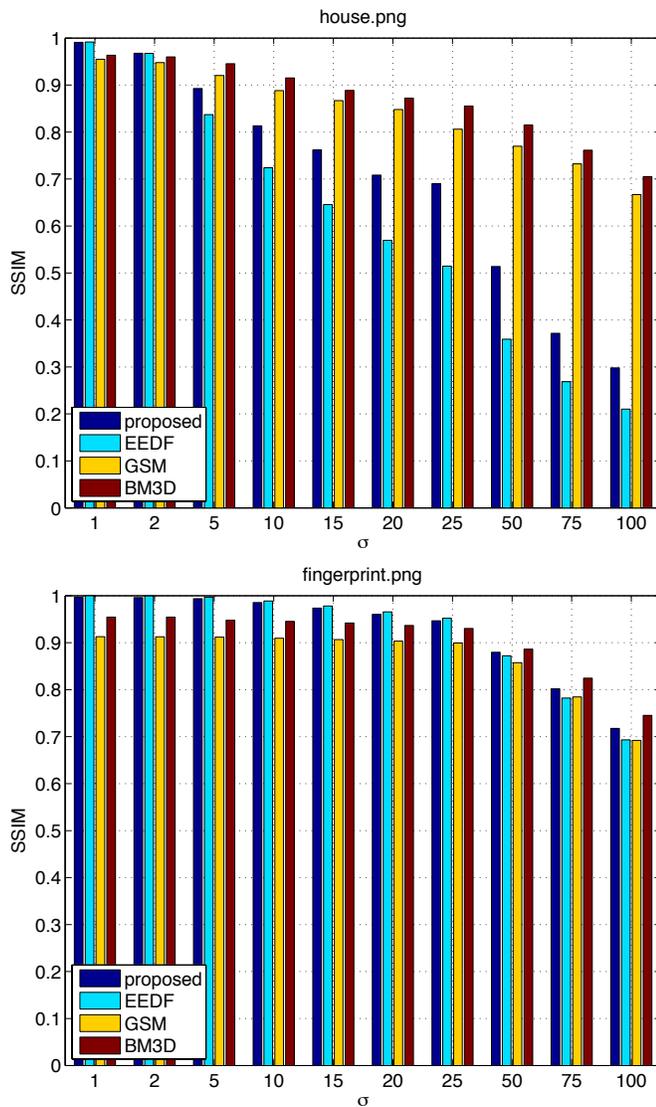


Fig. 4. SSIM results for the proposed method, EEDF, GSM, and BM3D for the house image and the fingerprint image.

TABLE II
VIF SCORES FOR THE PROPOSED METHOD

σ	Lena	Barb.	Boat	House	Pepper	Fingerp.	Flinst.
1	1.00	1.00	1.00	1.00	1.00	0.91	1.00
2	1.00	1.00	1.00	1.00	1.00	0.90	0.99
5	0.99	0.99	0.99	1.00	0.99	0.90	0.99
10	0.98	0.96	0.97	0.99	0.98	0.86	0.97
15	0.96	0.90	0.93	0.98	0.97	0.80	0.96
20	0.94	0.82	0.90	0.96	0.94	0.75	0.94
25	0.91	0.74	0.86	0.93	0.89	0.71	0.92
50	0.76	0.67	0.62	0.75	0.72	0.59	0.55
75	0.65	0.60	0.57	0.64	0.61	0.49	0.49
100	0.55	0.51	0.49	0.53	0.50	0.39	0.40

EEDF works only better for the fingerprint image, cf. Fig. 5 (bottom). The proposed method works as well as other recently published methods, see e.g., [16] in comparison to Table I.

Note that the quite obvious deviation from Portilla's original evaluation for the fingerprint image at low noise levels is due to using the estimated noise level instead of the added noise level

TABLE III
SSIM SCORES FOR THE PROPOSED METHOD

σ	Lena	Barb.	Boat	House	Pepper	Fingerp.	Flinst.
1	1.00	1.00	1.00	0.99	0.99	1.00	0.99
2	0.99	1.00	1.00	0.97	0.98	1.00	0.99
5	0.98	0.99	0.98	0.89	0.94	0.99	0.99
10	0.94	0.96	0.95	0.81	0.89	0.99	0.98
15	0.90	0.92	0.92	0.76	0.83	0.97	0.97
20	0.87	0.87	0.89	0.71	0.81	0.96	0.95
25	0.84	0.83	0.86	0.69	0.81	0.95	0.93
50	0.67	0.69	0.71	0.51	0.59	0.88	0.79
75	0.53	0.56	0.55	0.37	0.45	0.80	0.71
100	0.43	0.45	0.44	0.30	0.35	0.72	0.62

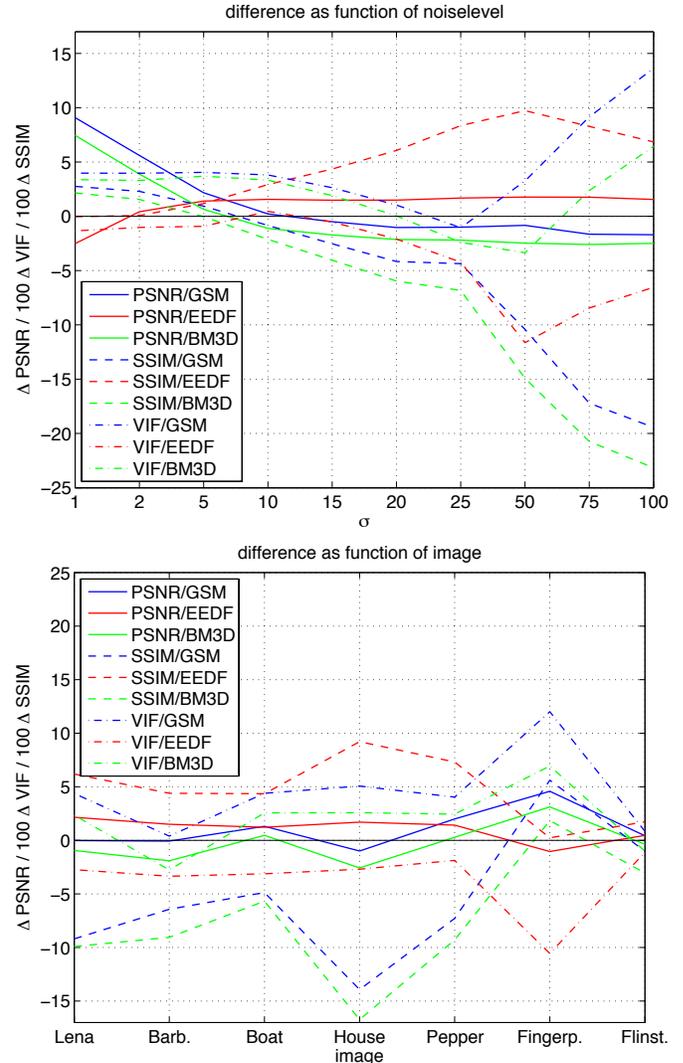


Fig. 5. Comparison of results as a function of noise (top) and as a function of the image (bottom). Positive values mean that the proposed method performs better than the competing method.

ground-truth. The fingerprint image contains a lot of noise even without added noise and hence the comparison for the PSNR is made to a very noisy image. If using the added noise level, the method reproduces the noise from the 'groundtruth' image. The obvious interpretation of this observation must be that GSM determines the amount of smoothing nearly entirely

from the noise level parameter, which also leads to low-pass effects, cf. also Fig. 1, bottom left. A similar dependency on the noise parameter is observed for BM3D. The proposed method is only mildly affected and EEDF is hardly affected by changing the noise level parameter for the fingerprint case.

As it can be seen from the VIF scores in Fig. 3 and the dash-dotted line plots in Fig. 5 (top) throughout all noise levels, the proposed method works better than GSM and worse than EEDF. BM3D is better than the proposed method in a very few cases. The same order of results is obtained if the results are integrated over the different noise levels, cf. Fig. 5 (bottom).

As it can be seen from the SSIM scores in Fig. 4 and the dashed line plots in Fig. 5 (top) for low noise levels, the proposed method works better than GSM and BM3D. For high noise levels, GSM and BM3D work better than the proposed method. The proposed method works better than EEDF throughout, but with an increasing difference with increasing noise level. Integrated over the different noise levels, GSM and BM3D work better than the proposed one on all images except for the fingerprint image and the proposed method works better than EEDF for all images, cf. Fig. 5 (bottom). The proposed method works better than the fields of expert except for the three highest noise levels, the house image, and medium to high level noise in the pepper image, cf. [16] in comparison to Table III.

The proposed method is significantly faster than GSM and EEDF (measured on a 3GHz Intel Core 2 Duo). Furthermore, it is extremely simple to implement and due to the low number of sequential steps in each iteration well suited for a GPU implementation. First performance evaluations on the GPU (own framework within DirectX on an Nvidia 8800GTX) allowed us to run 50 iterations in video real-time on PAL image size (720×576). 50 iterations are sufficient for all considered noise levels. The method can be made even faster by not updating the diffusion tensor in every iteration, but only in every second or third.

V. CONCLUSION

We proposed a new approach to define tensor driven diffusion based on the autocorrelation function. We show that our scheme is equivalent to a special case of adaptive filtering. We propose an optimized filter setup for the computation and compare the method to GSM, BM3D, and EEDF using three quality measures, PSNR, VIF, and SSIM. The result of the comparison is ambivalent in the way that the ranking of methods depends on the choice of the quality measure, the noise level, and the kind of image. Still, we observe that BM3D is always better than GSM and that our method is superior to GSM and BM3D for low-noise cases and images with many line structures. Compared to EEDF, the proposed method is clearly superior, except for the VIF scores. Visually, the diffusion results (EEDF and proposed method) contain fewer artifacts than the GSM results. The proposed scheme is easy to implement and fast to execute. Hence, we believe that our scheme is a good choice in many practical denoising tasks.

ACKNOWLEDGMENT

I would like to thank Dr. Hanno Scharr and Kai Krajsek for discussions on the topic and pointing me to references with relevant work. I further thank Johan Hedborg for running the evaluation on the GPU.

This research has received funding from the EC's 7th Framework Programme (FP7/2007-2013), grant agreements 21578 (DIPLECS) and 247947 (GARNICS), from ELLIIT, the Strategic Area for ICT research funded by the Swedish Government, and from the The Swedish Research Council through a grant for the project *Non-linear adaptive color image processing*.

REFERENCES

- [1] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [2] J. Weickert, "Theoretical foundations of anisotropic diffusion in image processing," *Computing, Suppl.*, vol. 11, pp. 221–236, 1996.
- [3] R. van den Boomgaard, "Nonlinear diffusion in computer vision," <http://staff.science.uva.nl/~rein/nldiffusionweb/material.html>, (Accessed 28 Jan 2008).
- [4] J. Weickert, K. Zuiderveld, B. ter Haar Romeny, and W. Niessen, "Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering," in *IEEE Int. Conf. on Image Processing*, 1997, pp. 396–399.
- [5] M. Welk, J. Weickert, and G. Steidl, "From tensor-driven diffusion to anisotropic wavelet shrinkage," in *European Conference on Computer Vision*, 2006.
- [6] G. H. Granlund and H. Knutsson, *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht, 1995.
- [7] J. Weule, "Iteration nichtlinearer Gauss-Filter in der Bildverarbeitung," Ph.D. dissertation, Heinrich-Heine-Universität Düsseldorf, 1994.
- [8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the 6th ICCV*, 1998.
- [9] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 17, no. 8, pp. 790–799, August 1995.
- [10] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [11] M. Felsberg and G. Granlund, "Anisotropic channel filtering," in *Proc. 13th Scandinavian Conference on Image Analysis*, ser. LNCS 2749, 2003, pp. 755–762.
- [12] J. Portilla, V. Strela, J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.
- [13] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, November 2006.
- [14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3d transform-domain collaborative filtering," *IEEE Trans. Image Processing*, vol. 16, no. 8, pp. 2080–2095, August 2007.
- [15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *International Conference on Computer Vision*, 2009, pp. 2272–2279.
- [16] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.
- [17] S. C. Zhu and D. Mumford, "Prior learning and Gibbs reaction-diffusion," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 19, no. 11, pp. 1236–1250, 1997.
- [18] M. Felsberg, P.-E. Forssén, and H. Scharr, "Channel smoothing: Efficient robust smoothing of low-level signal features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 209–222, 2006.
- [19] S. K. Weeratunga and C. Kamath, "A comparison of PDE-based nonlinear anisotropic diffusion techniques for image denoising," in *Image Processing: Algorithms and Systems II, SPIE Electronic Imaging*, 2003.
- [20] R. A. Carmona and S. Zhong, "Adaptive smoothing respecting feature directions," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 353–358, 1998.

- [21] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with PDE's : A common framework for different applications." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 506–517, 2005.
- [22] M. Felsberg, "On the relation between anisotropic diffusion and iterated adaptive filtering," in *30. DAGM Symposium Mustererkennung*, ser. LNCS, vol. 5096. Springer, 2008, pp. 436–445.
- [23] M. Felsberg, S. Kalkan, and N. Krüger, "Continuous dimensionality characterization of image structures," *Image and Vision Computing*, vol. 27, no. 6, pp. 628–636, 2009.
- [24] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.
- [25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [26] J. Bigün, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 775–790, August 1991.
- [27] W. Förstner, *Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung*, ser. C. Verlag der Bayerischen Akademie der Wissenschaften, 1991, no. 370.
- [28] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1965.
- [29] J. Bigün and G. H. Granlund, "Optimal orientation detection of linear symmetry," in *Proceedings of the IEEE First International Conference on Computer Vision*, London, Great Britain, June 1987, pp. 433–438.
- [30] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," in *ISPRS Intercommission Workshop, Interlaken*, June 1987, pp. 149–155.
- [31] J. Weickert, "A review of nonlinear diffusion filtering," in *Scale-Space Theory in Computer Vision*, ser. LNCS, B. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, Eds., vol. 1252. Springer, Berlin, 1997, pp. 260–271.
- [32] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Analysis and Machine Intell.*, vol. 13, no. 9, pp. 891–906, 1991.
- [33] H. Knutsson, R. Wilson, and G. H. Granlund, "Anisotropic non-stationary image estimation and its applications: Part I – restoration of noisy images," *IEEE Trans. on Communications*, vol. COM-31, no. 3, pp. 388–397, March 1983.
- [34] H. Knutsson, L. Haglund, and G. H. Granlund, "A new approach to image enhancement using tensor fields," in *Proc. PROART Workshop on Vision*, 1990, pp. 111–115.
- [35] B. Johansson, "Low level operations and learning in computer vision," Ph.D. dissertation, Linköping University, Sweden, SE-581 83 Linköping, Sweden, December 2004, dissertation No. 912, ISBN 91-85295-93-0.
- [36] H. Scharr, S. Körkel, and B. Jähne, "Numerische Isotropieoptimierung von FIR-Filtern mittels Querglättung," in *19. DAGM Symposium Mustererkennung, Braunschweig*, E.Paulus and F.M.Wahl, Eds., 1997, pp. 367–374.
- [37] J. Weickert, "Anisotropic diffusion in image processing," Ph.D. dissertation, Faculty of Mathematics, University of Kaiserslautern, 1996.
- [38] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.
- [39] P. D. Lax and R. D. Richtmyer, "Survey of the stability of linear finite difference equations," *Comm. Pure Appl. Math.*, vol. 9, pp. 267–293, 1956.
- [40] S. McKee and A. R. Mitchell, "Alternating direction methods for parabolic equations in two space dimensions with a mixed derivative," *The Computer Journal*, vol. 13, no. 1, pp. 81–86, 1970.
- [41] W. Förstner, "Image preprocessing for feature extraction in digital intensity, color and range images," in *Proc. Int'l. Summer School on Data Analysis and the Statistical Foundation of Geomatics*, ser. LNCS. Springer, 1998.
- [42] R. Terebes, O. Lavielle, P. Baylou, and M. Borda, "Mixed anisotropic diffusion," in *International Conference on Pattern Recognition*, vol. 3, 2002, pp. 760–763.
- [43] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 860–869.
- [44] J. Bai and X.-C. Feng, "Fractional-order anisotropic diffusion for image denoising," *IEEE Trans. Image Processing*, vol. 16, no. 10, pp. 2492–2502, 2007.



Michael Felsberg received the Diploma degree in 1998 and the PhD degree in 2002 (summa cum laude, awarded 'Fakultätspreis 2002') in engineering from Christian-Albrechts-University Kiel, Germany. During his PhD studies, he held scholarships from the DFG and the German National Merit Foundation. He became associate professor at Linköping University, Sweden, in 2004, docent in 2005, and full professor of computer vision and head of division in 2008. His research interests include multidimensional signal theory, image processing, low-level computer vision, and signal-processing based approaches to probabilistic representations and learning. He is coordinator of the Seventh Framework EU Programme Project DIPLECS. He has published more than 90 conference papers, journal articles, and book contributions on image processing, computer vision, and pattern recognition. He received paper awards from the German Pattern Recognition Society (DAGM) in 2000 and 2004 and from the Swedish Society for Automated Image Analysis (SSBA) in 2007 and 2010. He received the Olympus Award in 2005. He is vice-president of the SSBA.