

**Linköping University Post Print**

**On the Relation Between Anisotropic Diffusion  
and Iterated Adaptive Filtering**

Michael Felsberg

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

The original publication is available at [www.springerlink.com](http://www.springerlink.com):

Michael Felsberg, On the Relation Between Anisotropic Diffusion and Iterated Adaptive Filtering, 2008, Lecture Notes in Computer Science, 436-445.

Copyright: Springer-verlag

<http://www.springerlink.com/>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-44878>

# On the Relation Between Anisotropic Diffusion and Iterated Adaptive Filtering<sup>\*</sup>

Michael Felsberg

Computer Vision Laboratory, Linköping University  
S-58183 Linköping, Sweden, [mfe@isy.liu.se](mailto:mfe@isy.liu.se)

**Abstract.** In this paper we present a novel numerical approximation scheme for anisotropic diffusion which is at the same time a special case of iterated adaptive filtering. By assuming a sufficiently smooth diffusion tensor field, we simplify the divergence term and obtain an evolution equation that is computed from a scalar product of diffusion tensor and the Hessian. We propose further a set of filters to approximate the Hessian on a minimized spatial support. On standard benchmarks, the resulting method performs in average nearly as good as the best known denoising methods from the literature, although it is significantly faster and easier to implement. In a GPU implementation video real-time performance is achieved for moderate noise levels.

## 1 Introduction

Image denoising using non-linear diffusion [1] is a commonly used technique. 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 [3–5]. 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, many different 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], or multi-band filtering [12]. Optimal results usually require estimates of image priors [13, 14]. Some of the mentioned methods were related and compared in [15, 16]. The purpose of this paper is to introduce a novel, very simple numerical scheme for anisotropic diffusion, which

- is competitive to state of the art methods for image denoising
- relates anisotropic diffusion directly to iterated adaptive filtering
- is extremely simple to implement and real-time capable on GPU hardware.

---

<sup>\*</sup> This work has been supported by the CENIIT project CAIRIS.

The paper is structured as follows. In Sect. 2, we introduce some tensor notation and calculus and we give some background on anisotropic diffusion and adaptive filtering. In Sect. 3, 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 the choices of parameters. In Sect. 4, 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 drawbacks of the proposed method in the concluding Sect. 5.

## 2 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.

### 2.1 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:

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

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  $\lambda_i$ . Note that arbitrary powers of  $\mathbf{A}$  can be computed as

$$\begin{aligned} \mathbf{A}^n &= (\mathbf{O} \mathbf{\Lambda} \mathbf{O}^T)^n = \mathbf{O} \mathbf{\Lambda}^n \mathbf{O}^T = \mathbf{O} \text{diag}(\lambda_i^n) \mathbf{O}^T && \text{and due to linearity} \\ \exp(k \mathbf{A}) &= \mathbf{O} \text{diag}(\exp(k \lambda_i)) \mathbf{O}^T . \end{aligned} \quad (2)$$

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  $\nabla$ , such that

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

## 2.2 Anisotropic Diffusion

We will not go into much detail about the theory of anisotropic diffusion, the reader is referred to, e.g., the work of Weickert [17]. The diffusion scheme is defined as an evolution equation of an image  $b(\mathbf{x})$  over time  $t$  as:

$$\frac{\partial b}{\partial t} = \text{div}(\mathbf{D}(\mathbf{J})\nabla b) \quad (3)$$

where  $\mathbf{D}(\mathbf{J})$  is the diffusion tensor which is computed by modifying the eigenvalues of the structure tensor  $\mathbf{J}$ . The latter is computed by locally weighted averaging the outer product of the image gradient [18, 19]:

$$\mathbf{J}(\mathbf{x}) = \int w(\mathbf{y})\nabla b(\mathbf{x} - \mathbf{y})\nabla^T b(\mathbf{x} - \mathbf{y}) d\mathbf{y} . \quad (4)$$

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.

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. In the next step, the divergence is approximated by another numerical approximation of a derivative operator. Altogether, four numerical approximations of derivatives are made: the time-derivative (mostly by an explicit scheme), the image gradient, the image gradient used in the structure tensor (not necessarily the same) and the divergence. There are however exceptions, where semi-analytic solutions are implemented to reduce the number of numerical approximations [5].

## 2.3 Adaptive Filtering

Adaptive filtering is a more general and actually earlier published variant of steerable filters [20], developed by Knutson et al. [21, 22]. 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. The final formulation of the method can be found in [6], Chapt. 10. Adopting the notation above, the filter is composed as

$$h_{\text{adapt}} = h_{\text{LP}} + \sum_k \langle \mathbf{C}(\mathbf{J}) | \tilde{\mathbf{N}}_k \rangle h_{\text{HP},k} \quad (5)$$

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

$$H_{\text{HP},k} = (\mathbf{n}_k^T \mathbf{u})^2 \quad (6)$$

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

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

Finally, the rightmost term in (5) is proportional to the negative Hessian since

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

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

$$H_{\text{LP}} = 1 - \gamma R(|\mathbf{u}|) = 1 - \gamma |\mathbf{u}|^2 , \quad (7)$$

it is possible to show that, under some assumptions, the adaptive filtering implements a numerical scheme for anisotropic diffusion, see next section.

### 3 A Novel Numerical Approximation Scheme for Anisotropic Diffusion

In the first part of this section, we establish an equivalence between iterated adaptive filtering and a certain subset of numerical approximation schemes for anisotropic diffusion. We further propose a concrete scheme, which is used for the experiments further below and discuss the choices of parameters.

#### 3.1 On the Relation between Anisotropic Diffusion and Iterated Adaptive Filtering

The relation between anisotropic diffusion and adaptive filtering is based on the assumption that the divergence of the structure tensor vanishes. This can be justified since the structure tensor is obtained from local averaging, thus it is varying slowly. Hence, the approximation error that is made by neglecting the divergence of the tensor is small. We reconsider the anisotropic diffusion equation (3) and obtain with the product rule, the mentioned approximation, and (1)

$$\frac{\partial b}{\partial t} = \text{div}(\mathbf{D}(\mathbf{J})) \nabla b + \text{trace}(\mathbf{D}(\mathbf{J}) (\nabla \nabla^T b)) \approx 0 + \langle \mathbf{D}(\mathbf{J}) | \mathbf{H}b \rangle . \quad (8)$$

As it has been shown in the example in Sect. 2, the Hessian can be written in terms of a set of dual tensors and corresponding filters, such that a single iteration in the proposed anisotropic diffusion scheme

$$b_{t+1} = b_t + \gamma \langle \mathbf{D}(\mathbf{J}) | \mathbf{H}b \rangle \quad (9)$$

is equivalent to an adaptive filtering according to (5) with low-pass filter (7) (note that the example made use of the negative Hessian) and control tensor  $\mathbf{C} = \mathbf{Id} - \mathbf{D}$ . The time-discrete evolution (9) is the first step to establish a numerical scheme for anisotropic diffusion. The required spatial discretizations are developed in the next section.

### 3.2 A Novel Numerical Scheme for Anisotropic Diffusion

In order to implement (9) on a discrete grid, three filter (sets) have to be designed: The Hessian, the derivatives in (4), and the averaging kernel in (4). 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 (4) are implemented using Scharr filters [23], in order to obtain small filtermasks with good isotropy (note that on recent computer hardware it is counterproductive to implement such small filters as separable filters):

$$\frac{\partial b}{\partial x_1} \approx \frac{1}{32}([3 \ 10 \ 3]^T[-1 \ 0 \ 1]) * b \quad \frac{\partial b}{\partial x_2} \approx \frac{1}{32}([-1 \ 0 \ 1]^T[3 \ 10 \ 3]) * b .$$

Small filtermasks for gradients are preferable to large linear regularizers, as it has been shown in [24]. The previous two filter choices have comparably little influence on the quality of results – which is different for the design of the Hessian. Once again, the filtermask should be as little as possible. The smallest possible second order derivative approximation can be computed with a  $3 \times 3$  filter, which we will derive from one-sided finite differences. 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:

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

However, it shows to be preferable with respect to isotropy to also consider the corresponding 2D filter that averages in the vertical direction with a 2-box filter (see also [5]):

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

The actually used filters are the average of both approximations:

$$\frac{\partial^2}{\partial x_1^2} \approx \frac{1}{8} \begin{bmatrix} 1 & -2 & 1 \\ 6 & -12 & 6 \\ 1 & -2 & 1 \end{bmatrix} \quad \frac{\partial^2}{\partial x_2^2} \approx \frac{1}{8} \begin{bmatrix} 1 & 6 & 1 \\ -2 & -12 & -2 \\ 1 & 6 & 1 \end{bmatrix} \quad (10)$$

The corresponding mixed derivatives are fortunately the same for both derivative approximations and equal the one for the centered differences:

$$\frac{\partial^2}{\partial x_1 \partial x_2} \approx \frac{1}{4} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}^T = \frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} . \quad (11)$$

### 3.3 Useful Parameter Settings

Anisotropic diffusion has a fair amount of parameters. One of the most critical ones, the *stopping time*, is not considered in this paper. Same as in most denoising experiments, we report the best PSNR during the evolution. Preliminary experiments with a stopping time depending on the estimated noise level (see below), gave reasonable, but slightly worse results (about 0.3 dB). A second parameter that we have not investigated further is the *time step*  $\gamma$ : We use in all experiments  $\gamma = 0.05$  (small  $\gamma$  limits the time-discretization error), but the results do not change significantly under small variations of  $\gamma$ .

What we experienced as the most critical part of the algorithm in order to obtain comparative results, is 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 [14, 25]. The width of the influence function in robust estimation is much more important than the exact shape [25], such that we decided to choose an algebraically simple diffusivity of exponential form:

$$\mathbf{D}(\mathbf{J}) = \exp\left(-\frac{1}{k}\mathbf{J}\right). \quad (12)$$

Due to (2), this corresponds to applying the corresponding exponential mapping to the eigenvalues. Although it appears to be preferable to apply the exponential mapping directly on the matrix instead of an eigenvalue manipulation, the latter is faster since the eigenvalues can be extracted in closed form [5, 26]. In contrast to some diffusion schemes that combine explicitly isotropic and anisotropic diffusion [27], this is not necessary in our approach. We also apply the same scheme to edge-like images (e.g. the peppers image) and finger print images.

The most important parameter is the *relative width*  $k$  in (12). Based on our set of test images, however with different added noise, we empirically selected

$$k = 8 \cdot 10^{-6} (9 + \sigma_e)^2 \quad (13)$$

where  $\sigma_e$  is the *estimated noise* according to the method described in [28].

## 4 Experiments

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

### 4.1 Experimental Results

For the experiments, we used a popular set of images that has already been used in earlier publications [12, 13, 29]<sup>1</sup>. The evaluation criteria is the PSNR. The still most successful method in terms of PSNR is published by Portilla et al. [12], so that we compare our method with their implementation. We evaluate

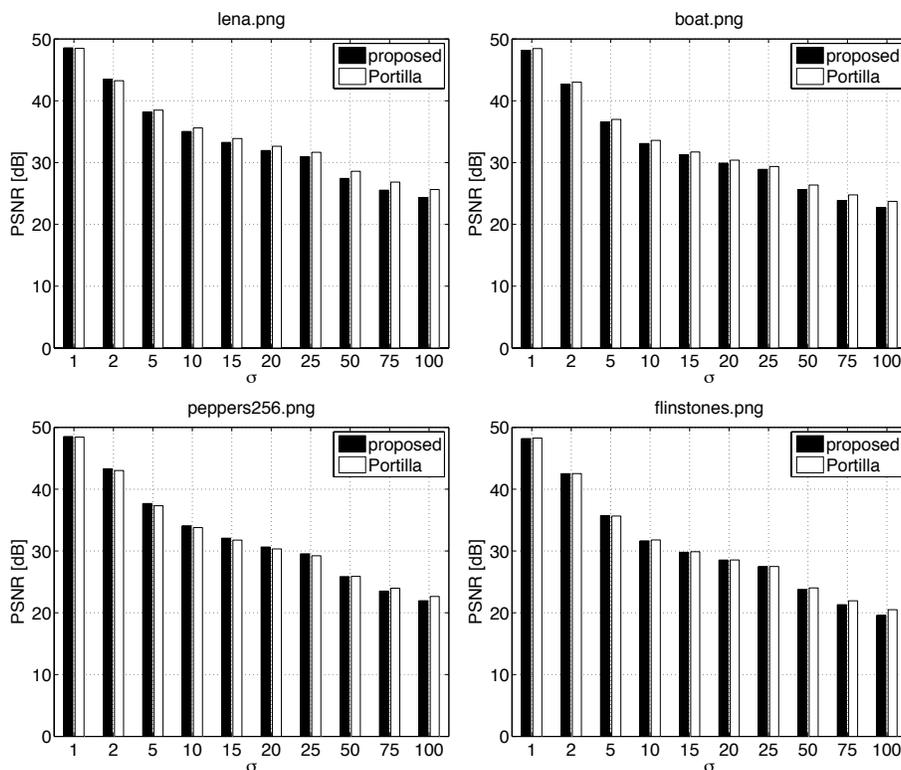
<sup>1</sup> Where the authors in [29] miss the fact that they are actually using a DCT as mentioned in their reference [20].

each image and noise level with ten different instances of noise. Our method is entirely implemented in Matlab and took about 2.5 hours on a 1.66 GHz Intel Core Duo. Portilla's method is a combination of Matlab and C code and took more than 12 hours on the same machine.

The resulting PSNRs for both methods are plotted in Fig. 1 for a selection of images. Since the PSNR does not reflect what terms of artifacts are introduced, we also show the resulting image in one case where both methods performed equally well (see Fig. 2). Finally, we also show the standard deviation of the PSNR for both methods as a function of the noise level.

## 4.2 Discussion of Results

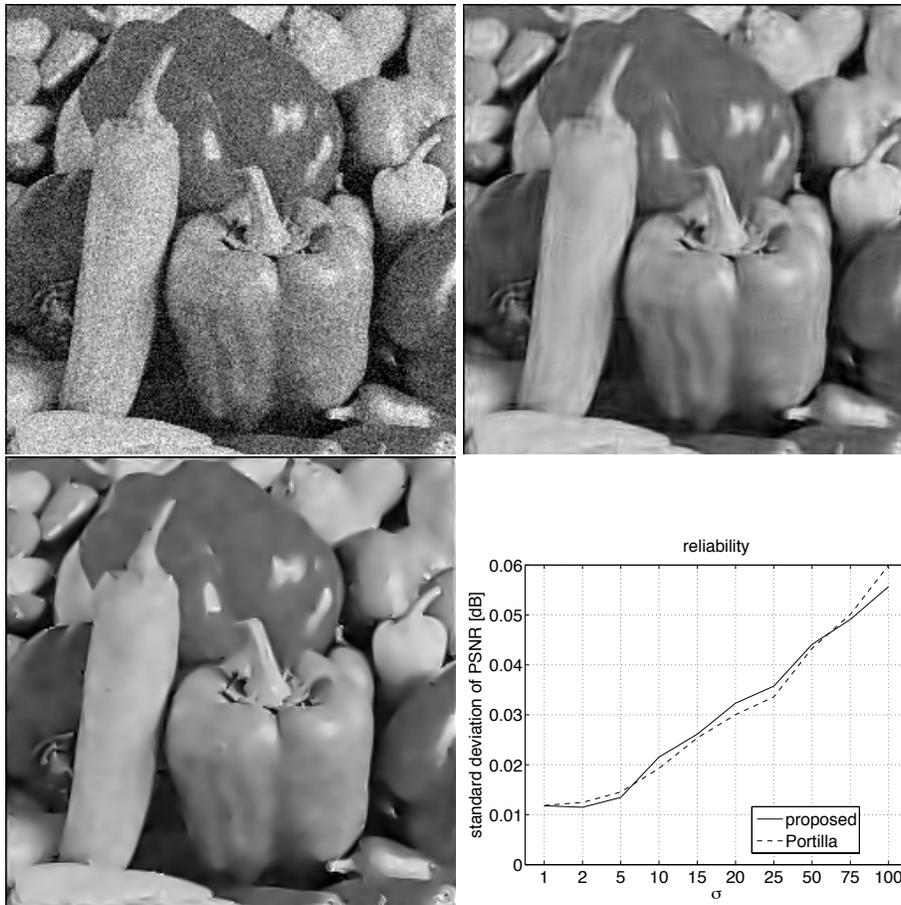
As it can be seen from the PSNRs in Fig. 1, the proposed method works in average nearly as good as the method of Portilla, in certain cases it is even better. In nearly all cases it works better than other recently published results, e.g., [13]. The remaining errors in the denoised result have however entirely



**Fig. 1.** Experimental results for the proposed method and Portilla's method for four different images.

different appearance. As shown in Fig. 2, the method by Portilla introduces serious oscillation artifacts in contrast to the proposed method. The variance of the PSNR results is neglectable in both cases.

The proposed method is significantly faster than Portilla's method. 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) allowed us to run 50 iterations in video real-time on PAL image size ( $720 \times 576$ ). 50 iterations are sufficient for noise levels up to  $\sigma = 25$  (see also Fig. 2), i.e., in most



**Fig. 2.** Experimental results part two, row-wise from left to right: The peppers image with heavy noise ( $\sigma = 25$ ), the result from Portilla's method, the result of the proposed method, and the standard deviation of the PSNRs.

practical relevant cases. The method can be made even faster by not updating the diffusion tensor in every iteration, but only in every second or third.

## 5 Conclusion

We consider the proposed scheme for anisotropic diffusion to be on the same level as state of the art image denoising methods concerning PSNR. Visually, the results contain fewer artifacts and the scheme is real-time capable. Hence, we believe that our scheme is a good choice in many practical denoising tasks. What remains for future work is to analyze the asymptotic stability of the scheme, to further improve isotropy, to evaluate other quantitative measures, and finally to try to find upper bounds for the approximation error caused by neglecting the divergence of the diffusion tensor.

## Acknowledgment

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

## References

1. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence* **12**(7) (1990) 629–639
2. Weickert, J.: Theoretical foundations of anisotropic diffusion in image processing. *Computing, Suppl.* **11** (1996) 221–236
3. v. d. Boomgaard, R.: Nonlinear diffusion in computer vision. <http://staff.science.uva.nl/~rein/nldiffusionweb/material.html> (Accessed 28 Jan 2008).
4. Weickert, J., Zuiderveld, K., ter Haar Romeny, B., Niessen, W.: Parallel implementations of AOS schemes: A fast way of nonlinear diffusion filtering. In: *IEEE Int. Conf. on Image Processing*. (1997) 396–399
5. Welk, M., Weickert, J., Steidl, G.: From tensor-driven diffusion to anisotropic wavelet shrinkage. In: *European Conference on Computer Vision*. (2006)
6. Granlund, G.H., Knutsson, H.: *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht (1995)
7. Weule, J.: *Iteration nichtlinearer Gauss-Filter in der Bildverarbeitung*. PhD thesis, Heinrich-Heine-Universität Düsseldorf (1994)
8. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Proceedings of the 6th ICCV*. (1998)
9. Cheng, Y.: Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Analysis and Machine Intell.* **17**(8) (August 1995) 790–799
10. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intell.* **24**(5) (2002) 603–619
11. Felsberg, M., Granlund, G.: Anisotropic channel filtering. In: *Proc. 13th Scandinavian Conference on Image Analysis. LNCS 2749* (2003) 755–762

12. Portilla, J., Strela, V., Wainwright, J., Simoncelli, E.P.: Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Processing* **12**(11) (2003) 1338–1351
13. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: *Computer Vision and Pattern Recognition*. Volume 2. (2005) 860–869
14. Zhu, S.C., Mumford, D.: Prior learning and Gibbs reaction-diffusion. *IEEE Trans. Pattern Analysis and Machine Intell.* **19**(11) (1997) 1236–1250
15. Felsberg, M., Forssén, P.E., Schar, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(2) (2006) 209–222
16. Weeratunga, S.K., Kamath, C.: A comparison of PDE-based non-linear anisotropic diffusion techniques for image denoising. In: *Image Processing: Algorithms and Systems II*, SPIE Electronic Imaging. (2003)
17. Weickert, J.: A review of nonlinear diffusion filtering. In ter Haar Romeny, B., Florack, L., Koenderink, J., Viergever, M., eds.: *Scale-Space Theory in Computer Vision*. Volume 1252 of LNCS., Springer, Berlin (1997) 260–271
18. Bigün, J., Granlund, G.H.: Optimal orientation detection of linear symmetry. In: *Proceedings of the IEEE First International Conference on Computer Vision*, London, Great Britain (June 1987) 433–438
19. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centres of circular features. In: *ISPRS Intercommission Workshop*, Interlaken. (June 1987) 149–155
20. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intell.* **13**(9) (1991) 891–906
21. Knutsson, H., Wilson, R., Granlund, G.H.: Anisotropic non-stationary image estimation and its applications: Part I – restoration of noisy images. *IEEE Trans. on Communications* **COM-31**(3) (March 1983) 388–397
22. Knutsson, H., Haglund, L., Granlund, G.H.: A new approach to image enhancement using tensor fields. In: *Proc. PROART Workshop on Vision*. (1990) 111–115
23. Schar, H., Körkel, S., Jähne, B.: Numerische Isotropieoptimierung von FIR-Filtern mittels Querglättung. In E.Paulus, F.M.Wahl, eds.: *19. DAGM Symposium Mustererkennung*, Braunschweig. (1997) 367–374
24. Johansson, B.: *Low Level Operations and Learning in Computer Vision*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden (December 2004) Dissertation No. 912, ISBN 91-85295-93-0.
25. Black, M.J., Rangarajan, A.: On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision* **19**(1) (1996) 57–91
26. Bigün, J., Granlund, G.H., Wiklund, J.: Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(8) (August 1991) 775–790
27. Terebes, R., Lavialle, O., Baylou, P., Borda, M.: Mixed anisotropic diffusion. In: *International Conference on Pattern Recognition*. Volume 3. (2002) 760–763
28. Förstner, W.: 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*. LNES, Springer (1998)
29. Bai, J., Feng, X.C.: Fractional-order anisotropic diffusion for image denoising. *IEEE Trans. Image Processing* **16**(10) (2007) 2492–2502