Rock image denoising by fast bilateral filtering

Giuseppe Papari, Nasiru Idowu, Trond Varslot

FEI Lithicon Digital Rock Services
Stiklestadveien 1, N-7041 Trondheim, Norway

Abstract. A fast denoising algorithm for 3D μCT images is proposed. The algorithm is based on an efficient implementation of bilateral filtering, in which the filter kernel is decomposed into a series of cosines. This reduces the evaluation of bilateral filtering to a Gaussian convolutions, for which very fast recursive algorithms are available. The frequencies of the cosine terms are optimized in the minimal square error sense and an analytical expression of the corresponding coefficients is provided. A detailed computational analysis of the proposed method is presented as well. The method is validated with images that are encountered in digital rock applications. Comparison with state of the art existing methods show the superiority of the proposed algorithm in terms of noise rejection and computation time.

1 Introduction

Digital rock physics, namely the measurement of several macroscopic physical properties of a rock from 3D scans of samples, is an applicative area in which image processing has been increasing its importance in the last decades [1, 2]. A common pipeline consists in segmenting X-ray scans of rock samples and extract geometrical models of their solid and pore spaces to perform numerical simulations of several physical phenomena, such as multiphase fluid flow, wave propagation, electric flow, and many others. These simulations enable to estimate several petrophysical properties which would be difficult to measure accurately in a standard experimental setting in which the rock samples may be damaged. [3, 4].

The initial step of this process is to acquire and segment 3D X-ray scans of rock samples. However, the quality of the segmentation may be compromised by the noise that is always present in high resolution μCT scans due to limited acquisition time [5], with an obvious negative impact on the rest of the pipeline. Therefore, a preprocessing step aimed at reducing the noise level is mandatory.

Processing the images that are encountered in digital rock applications offers several peculiar challenges. First of all, capturing the geometry of the pore space of a rock sample with the needed accuracy results in very large images; specifically, the typical spatial extent of a rock sample is in the order of several millimeters in each dimension, and a common resolution is $2\mu m$ per voxel, thus yielding image in the size of $10^9 - -10^{11}$ voxels per sample. This imposes severe restrictions on the design of a denoising algorithm in terms of amount

of used memory, number of calculations per voxel, and scalability with respect to parallelization. In addition, for many digital rock applications the knowledge of the size of the *narrowest* throats and corners in the pore space is the most crucial. Therefore, it is important to have denoising techniques which not only smooth out noise while keeping edges sharp, but also preserve corners and narrow elongated elements in the input image.

For these reasons, not every available denoising algorithm is suitable for digital rock imaging. For example, filters based on variational methods [6] and partial differential equation [7, 8], which are powerful and versatile, cannot be afforded in this context because of their iterative nature, which makes the computation time quite high.

In this paper, a non-iterative denoising algorithm is proposed, which is based on bilateral filtering (BF) [9, 10]. The output of BF is a weighted average of the the input graylevels on a neighborhood of each voxel, where the weights are determined adaptively to minimize the amount of smoothing across edges, corners, and other elongated elements. A fast implementation of BF is presented, which consists in expanding the kernel of the filter into a sum of cosines, thus reducing BF to a few Gaussian convolutions. the frequencies of the cosines are optimized in a minimum square error (MSE) sense, thus minimizing the number of terms in the expansion that are needed for convergence.

The proposed method is validated with experimental results related to μCT scans of rock samples and a comparison with existing nonlinear filters is made in terms of both denoising performance and computation time.

2 Proposed method

In this section, the proposed denoising algorithm is presented, which is based on fast BF. A quick overview of the proposed technique is given in Subsection 2.1, while the fast algorithm for BF is detailed in Subsection 2.2.

2.1 Overview

The proposed filtering method is illustrated in Fig. 1. The first step is histogram equalization. Specifically, the histogram $p(I)$ of the input image $I(\mathbf{r})$ is estimated along with the cumulative histogram $F(I)$, which is defined as:

$$F(I) \triangleq \int_0^I p(x)\mathrm{d}x \tag{1}$$

Then, the following image $I_H(\mathbf{r})$ is evaluated by means of a zero-memory nonlinear function:

$$I_H(\mathbf{r}) \triangleq F[I(\mathbf{r})] \tag{2}$$

It is easy to prove that the image $I_H(\mathbf{r})$ has the same morphology of the input image $I(\mathbf{r})$, but with a uniformly distributed graylevel histogram.

The second step is BF, for which a fast algorithm is presented in the next subsection. Bilateral filtering is the core of the denoising method, as it removes noise while keeping edges sharp. Finally, the graylevel distortion (2) is compensated by the application of the inverse function F^{-1} to $U_H(\mathbf{r})$:

$$U(\mathbf{r}) = F^{-1}[U_H(\mathbf{r})]. \tag{3}$$

The reason of this pre and post processing, instead of applying BF directly on the input image will be clear at the end of the next subsection.

Fig. 1. Overview of the proposed denoising algorithm.

2.2 Fast bilateral filtering

Let $I(\mathbf{r})$ be a luminance profile defined on a discrete n-dimensional domain $\Omega \subseteq \mathbb{Z}^n$, with $\mathbf{r} \in \Omega$. Without loss of generality, it will be assumed through the whole paper that the values of $I(\mathbf{r})$ range between 0 and 1. The output $U(\mathbf{r})$ of the BF is defined as follows:

$$U(\mathbf{r}) = \frac{N(\mathbf{r})}{D(\mathbf{r})} \tag{4}$$

with:

$$\begin{aligned} N(\mathbf{r}) &= \sum_{\boldsymbol{\rho} \in \Omega} I(\boldsymbol{\rho}) g_{\sigma_1}(\mathbf{r} - \boldsymbol{\rho}) w_{\sigma_2}[I(\mathbf{r}) - I(\boldsymbol{\rho})] \\ D(\mathbf{r}) &= \sum_{\boldsymbol{\rho} \in \Omega} g_{\sigma_1}(\mathbf{r} - \boldsymbol{\rho}) w_{\sigma_2}[I(\mathbf{r}) - I(\boldsymbol{\rho})] \end{aligned} \tag{5}$$

where $\sigma_1 > 0$ and $\sigma_2 > 0$ are scale parameters, and the weigthing functions $g_\sigma(\mathbf{u})$ and $w_\sigma(u)$ are both Gaussian in the great majority of applications.

As we see, the output of BF on point $\mathbf{r}$ is a weighted average of graylevels of the input image on a neighborhood of $\mathbf{r}$, where the weights are decreasing functions both with the spatial and the range distance from the central point. The presence of the range terms w_{σ_2} guarantees that whenever $\mathbf{r}$ is close to an edge only those points $\boldsymbol{\rho}$ that are on the same side of the edge give a significant contribution to the sums (5). Therefore, no significant smoothing is done across edges. In a similar way, corners and other narrow structure in the images are preserved as well.

The direct implementation of BF through its definition (4) and (5) is computationally infeasible for 3D images, as its computational complexity is *cubic* with the scale parameter σ_1. In contrast, a faster algorithm is proposed here.

Specifically, it is assumed that the range distance $w_{\sigma_2}(u-v)$ admits the following expansion:

$$w_\sigma(u-v) = \sum_{k=1}^{N} c_k(\sigma)\psi_k(u)\psi_k(v), \tag{6}$$

where the functions $\psi_k(u)$ are for the moment assumed to be given. By plugging (6) into (4), we get the following approximation $\hat{U}(\mathbf{r})$ of $U(\mathbf{r})$:

$$\hat{U}(\mathbf{r}) = \frac{\hat{N}(\mathbf{r})}{\hat{D}(\mathbf{r})} \tag{7}$$

with:

$$\begin{aligned} \hat{N}(\mathbf{r}) &= \sum_{k=1}^{N} c_k(\sigma)\psi_k[I(\mathbf{r})] \sum_{\boldsymbol{\rho}\in\Omega} g_{\sigma_1}(\mathbf{r}-\boldsymbol{\rho})I(\boldsymbol{\rho})\psi_k[I(\boldsymbol{\rho})] \\ \hat{D}(\mathbf{r}) &= \sum_{k=1}^{N} c_k(\sigma)\psi_k[I(\mathbf{r})] \sum_{\boldsymbol{\rho}\in\Omega} g_{\sigma_1}(\mathbf{r}-\boldsymbol{\rho})\psi_k[I(\boldsymbol{\rho})]. \end{aligned} \tag{8}$$

Both sums in $\boldsymbol{\rho}$ are convolution with a Gaussian mask, thus reducing the computation of the bilateral filter to evaluating $2N$ convolutions. The whole computational flow of this algorithm is illustrated in Fig. 2. This is much faster than evaluating (6) directly in the spatial domain, due to very fast available algorithms which compute convolutions [11, 12] with the needed accuracy. In particular, their computational complexity is linear with the size of the image, and *constant* with respect to σ_1, instead of cubic.

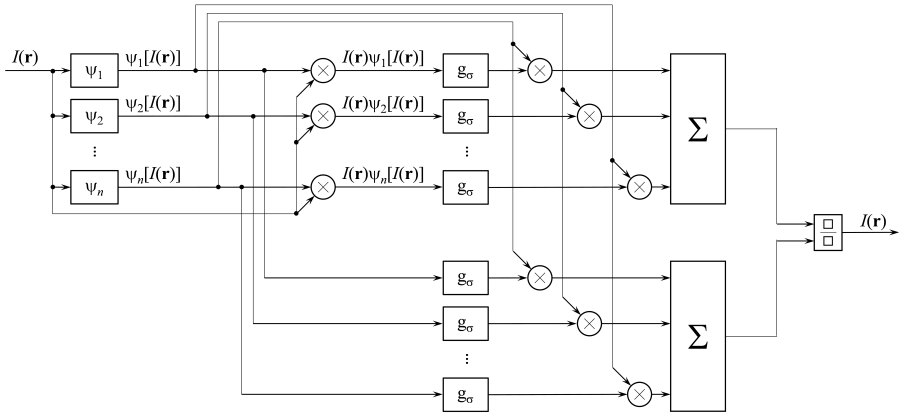

Fig. 2. Computation scheme of bilateral filtering.

In order to determine the functions $\psi_k(u)$, we follow the Lie-theoretical approach described in [13]. Specifically, the following condition is imposed:

$$\psi_k(u-v) = \psi_k(u)\phi_k(v), \tag{9}$$

where $\phi_k(v)$ are arbitrary functions. Without loss of generality, the functions $\psi_k(u)$ are also assumed to be orthogonal:

$$\int_0^1 \psi_k(u)\psi_s(u)\mathrm{d}u = 0, \qquad k \neq s \tag{10}$$

It is easy to show that the only functions that satisfy (9) and (10) are $\psi_k(u) = \exp(i\omega_k u)$ and $\phi_k(u) = \exp(-i\omega_k u)$, as detailed in [13]. This results in the following Fourier expansion of the range term $w_\sigma(u)$, which contains only cosine terms only:

$$w_\sigma(u) = \sum_k c_k(\sigma)\cos(\omega_k u) \tag{11}$$

where the values of ω_k are for the moment arbitrary. This yields:

$$\begin{aligned} w_\sigma(u-v) &= \sum_k c_k(\sigma)\cos(\omega_k u - \omega_k u) = \\ &= \sum_k c_k(\sigma)[\cos(\omega_k u)\cos(\omega_k v) + \sin(\omega_k u)\sin(\omega_k v)] \end{aligned} \tag{12}$$

By comparing (12) with (6) we see that the requested functions $\psi_k(u)$ are cosines and sines in u. As to the coefficients $c_k(\sigma)$, they are equal to:

$$\begin{aligned} c_k(\sigma) &= \frac{\displaystyle\int_0^1\int_0^1 w_\sigma(u-v)\cos[\omega_k(u-v)]\mathrm{d}u\mathrm{d}v}{\displaystyle\int_0^1\int_0^1 \cos[\omega_k(u-v)]\mathrm{d}u\mathrm{d}v} = \\ &= \frac{\displaystyle\int_0^1 w_\sigma(u)(1-u)\cos(\omega_k u)\mathrm{d}u}{\displaystyle\int_0^1 (1-u)\cos^2(\omega_k u)\mathrm{d}u} = \\ &= \frac{4}{1+\mathrm{sinc}^2(\omega_k)}\int_0^1 w_\sigma(u)(1-u)\cos(\omega_k u)\mathrm{d}u \end{aligned} \tag{13}$$

with $\mathrm{sinc}(u) \triangleq \frac{\sin(u)}{u}$.

The last integral $\eta(\sigma,\omega_k) \triangleq \int_0^1 w_\sigma(u)(1-u)\cos(\omega_k u)\mathrm{d}u$ can be expressed analytically for the Gaussian case $w_\sigma(u) = \exp(-\frac{u^2}{2\sigma^2})$ in terms of complex error functions. Specifically, a laborious calculation yields:

$$\begin{aligned} \eta(\sigma,\omega) = \frac{\sigma}{4}\Bigg\{ & \sqrt{2\pi}e^{-\frac{\sigma^2\omega^2}{2}}\Phi(\sigma,\omega) \\ & - \sigma\Bigg[4 - 4e^{-\frac{1}{2\sigma^2}}\cos(\omega) - 2\sqrt{2\pi}e^{-\frac{\sigma^2\omega^2}{2}}\,\mathrm{erfi}\left(\frac{\sigma\omega}{\sqrt{2}}\right) \\ & + \sqrt{2\pi}e^{-\frac{\sigma^2\omega^2}{2}}\Psi(\sigma,\omega)\Bigg]\Bigg\} \end{aligned} \tag{14}$$

with:

$$\begin{aligned}\Phi(\sigma,\omega) &\triangleq \operatorname{erf}\left(\frac{\sigma^{-1}+i\sigma\omega}{\sqrt{2}}\right)+\operatorname{erf}\left(\frac{\sigma^{-1}-i\sigma\omega}{\sqrt{2}}\right)\\ \Psi(\sigma,\omega) &\triangleq \operatorname{erfi}\left(\frac{\sigma\omega+i\sigma^{-1}}{\sqrt{2}}\right)+\operatorname{erfi}\left(\frac{\sigma\omega-i\sigma^{-1}}{\sqrt{2}}\right)\end{aligned} \tag{15}$$

and $\operatorname{erfi}(z) \triangleq -i\operatorname{erf}(iz)$, where $\operatorname{erf}(x)$ is the standard error function $\operatorname{erf}(x) \triangleq \int_0^x e^{-u^2}\mathrm{d}u$. The function $\eta(\sigma,\omega)$ is plotted vs ω in Fig. 3 for different values of σ. As we see, the smaller σ is, the more slowly $\eta(\sigma,\omega)$ goes to zero. This indicates that the number of non-negligible coefficients increases as σ decreases.

Finally, the values of ω_k are computed. By plugging $\psi_k(u) = \cos(\omega_k u)$ into the orthogonality condition (10), we get:

$$\frac{\tan(\omega_k)}{\omega_k} = \frac{\tan(\omega_s)}{\omega_s}, \qquad \forall k,s = 1,2,... \tag{16}$$

This trascendent equation is easy to solve numerically and allows expressing ω_k as a function of ω_0, *i.e.*, $\omega_k = \omega_k(\omega_0)$, with $\omega_k \in \left[\left(k-\frac{1}{2}\right)\pi, \left(k+\frac{1}{2}\right)\pi\right)$. The functions $\omega_k(\omega_0)$ are plotted in Fig. 4 for $\omega_0 \in [0,\pi/2]$ for $k = 1,...,6$.

The value of ω_0 is obtained by minimizing the following quadratic error $\epsilon_\sigma^2(\omega_0)$:

$$\begin{aligned}\epsilon_\sigma^2(\omega_0) &\triangleq \frac{1}{2}\int_0^1\int_0^1\left\{w_\sigma(u-v)-\sum_{k=1}^{N}c_k(\sigma,\omega_0)\cos[\omega_k(\omega_0)(u-v)]\right\}^2\mathrm{d}u\mathrm{d}v =\\ &= \int_0^1\left\{w_\sigma(u)-\sum_{k=1}^{N}c_k(\sigma,\omega_0)\cos[\omega_k(\omega_0)u]\right\}^2(1-u)\mathrm{d}u,\end{aligned} \tag{17}$$

where $c_k(\sigma,\omega_0)$ is the value of (13) with $\omega_k = \omega_k(\omega_0)$.

Summarizing, BF is reduced to 2N convolutions as illustrated in Fig. 2, by means of the decomposition (6). The functions $\psi_k(u)$ are cosines and sines, where the frequencies ω_k are determined by minimizing the MSE (17). This minimization is reduced to a 1D problem by means of the orthogonality relations (16) and can be precomputed off line. After evaluating the ω_k, the coefficients $c_k(\sigma)$ are also computed off line by evaluating the integral (13).

It is easy to prove that the minimization of the MSE (17) is optimal for an input image which has a uniformly distributed graylevel histogram, since all values of u and v contribute with the same weight. Therefore, the input image is pre-distorted as in (2) so that BF is applied to an image with uniformly distributed histogram, and the inverse distortion is applied after filtering. This results in a smaller number of convolutions compared to applying the BF directly to the input image.

The whole computational complexity of the proposed algorithm is detailed in Table 1, as can be easily inferred from Fig. 2, where N is the number of

terms of the expansion (6) and N_V is the number of voxels in the input image. Each convolution is evaluated by means of the algorithm proposed in [11] which requires, in 3D, $30N_V$ floating point multiplications and additions. Therefore, the overall computational complexity is roughly $60NN_V$, which is linear in the size of the image and constant in σ_1. In practice the number of convolutions that are needed for a good accuracy is less than 10 in the great majority of situations, thus keeping the number of floating point operations below $600N_V$.

On the other hand, the direct implementation of (4) and (5) would require to loop, for each voxel, on a cube of radius 3σ, thus requiring $2 \cdot (7\sigma_1)^3 N_V > 600\sigma_1^3 N_V$ floating point multiplications and the same number of additions. For the relatively small value $\sigma = 3$ the number of additions and multiplications per voxel is around 10^5, while it is around $10^2 - -10^3$ for the proposed method.

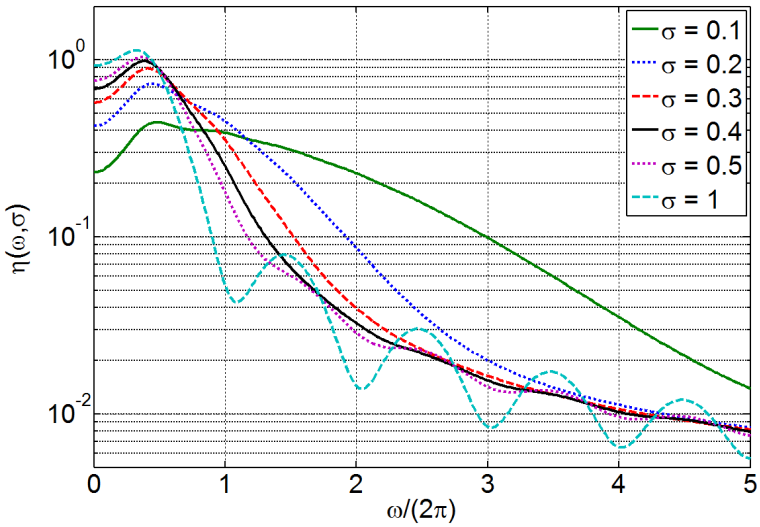

Fig. 3. Plots of the function $\eta(\sigma, \omega)$ vs ω for several values of σ.

Table 1. Computational complexity of the proposed BF algorithm

$2N$	Gaussian convolutions
NN_V	Nonlinear function evaluations
$(3N+1)N_V$	Floating point multiplications and divisions
$2(N-1)N_V$	Floating point additions

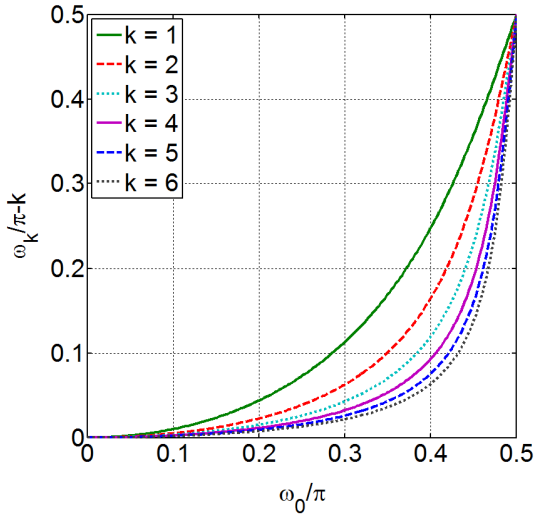

Fig. 4. Plots of the solutions $\omega_k(\omega_0)$ of (16) for several values of k.

3 Experimental results

The proposed method has been applied to several 3D μCT scans of sandstone rocks. The size of each input image is 800 voxels in each dimension and the resolution is $2\mu m$ per voxel. For all test images, the method has been applied with the same values of the input parameters, which are $\sigma_1 = 5$ and $\sigma_2 = 0.2$. The proposed method has also been compared with the nonlinear diffusion algorithm proposed in [14], for which the input parameters have been selected as suggested in [14].

Examples of outputs of the studied techniques are shown in Figs. 5-6, in which crops of slices on the xy plane are displayed for better visualization purposes. We observe that in all studied cases the proposed algorithm effectively removes noise while preserving edges. Conversely, the anisotropic diffusion algorithm removes less noise.

We also notice that the residual noise of the proposed method has a rather low spatial correlation. This makes it easy to further reduce the noise level by means of a second application of the filter, as illustrated in Fig. 7. This is not the case for anisotropic diffusion. In particular, it has a more highly correlated residual noise, which is harder to attenuate without introuducing a certain amount of undesired smoothing.

Finally, a comparison on the computation time is presented. Specifically, in Table 2 the average computation time is reported along with the standard deviation across all test images for the proposed algorithm, the implementation of BF available in imageJ (susan filter) [15], and the anisotropic diffusion algorithm presented in [14]. As we see, the proposed algorithm is over ten times faster than both anisotropic and the implementation of BF provided with ImageJ. Though

these timings are indicative, since they depend on how well each algorithm has been optimized, they clearly show the superiority of the proposed approach over the other considered techniques.

Table 2. Computation time of the studied algorithms.

Algorithm	Average time	Std. dev.
Proposed algorithm	1427.1s	25.3s
ImageJ BF (susan)	19643s	138.4s
Nonlinear diffuision	19093s	148.6

4 Summary, discussion and conclusions

A fast denoising algorithm for 3D μCT rock images has been presented, which is based on BF. Specifically, the range term $w_{\sigma_2}(u)$ has been expanded into a Fourier series, thus reducing BF to the calculation of few Gaussian convolutions. This is much faster than implementing BF from its definition (4) and (5) directly, due to very efficient available algorithms for Gaussian filtering [11, 12].

Reducing BF to several convolutions by means of an expansion in the form (6) had been proposed in earlier work too [16, 17]. However, non-optimal expansions were considered, which converge very slowly resulting in a slower filter. Specifically, a Taylor expansion of $w_{\sigma_2}(u)$ is considered in [16]. However, it has a good convergence only for very small values of u and requires the evaluation of high-order polynomials, leading to numerical instability. An attempt to lessen this problem has been proposed in [17], in which trigonometric polynomials are used instead of Taylor series. However, their expansion still favors small values of u, resulting in a slow convergence. In particular, the number of terms that are needed for a given accuracy grows *quadratically* with $\frac{1}{\sigma_2}$.

In contrast, the optimal design proposed in this paper results in a much faster convergence. More specifically, the values of ω_k obtained by minimizing the MSE (17) are optimal for the case of uniformly distributed histogram of the input image. Since in many situations this not the case, a pre and post distortion of the image are performed, so that BF is applied to an image with uniform histogram. This results in a less number of coefficients, compared to applying BF directly to the input image without compromising accuracy.

A possible alternative could be to modify the cost function (17) so that the values of ω_k are optimized for the specific image histogram. However, this would require to recalculate the values of ω_k for every new image to filter. In contrast, in the proposed method the optimization of the ω_k is done off line, thus making the filter faster.

To summarize, the denoising algorithm proposed here is fast and effective in filtering μCT images that are encountered in digital rock applications. Its performance are higher compared to existing algorithms, such as anisotropic

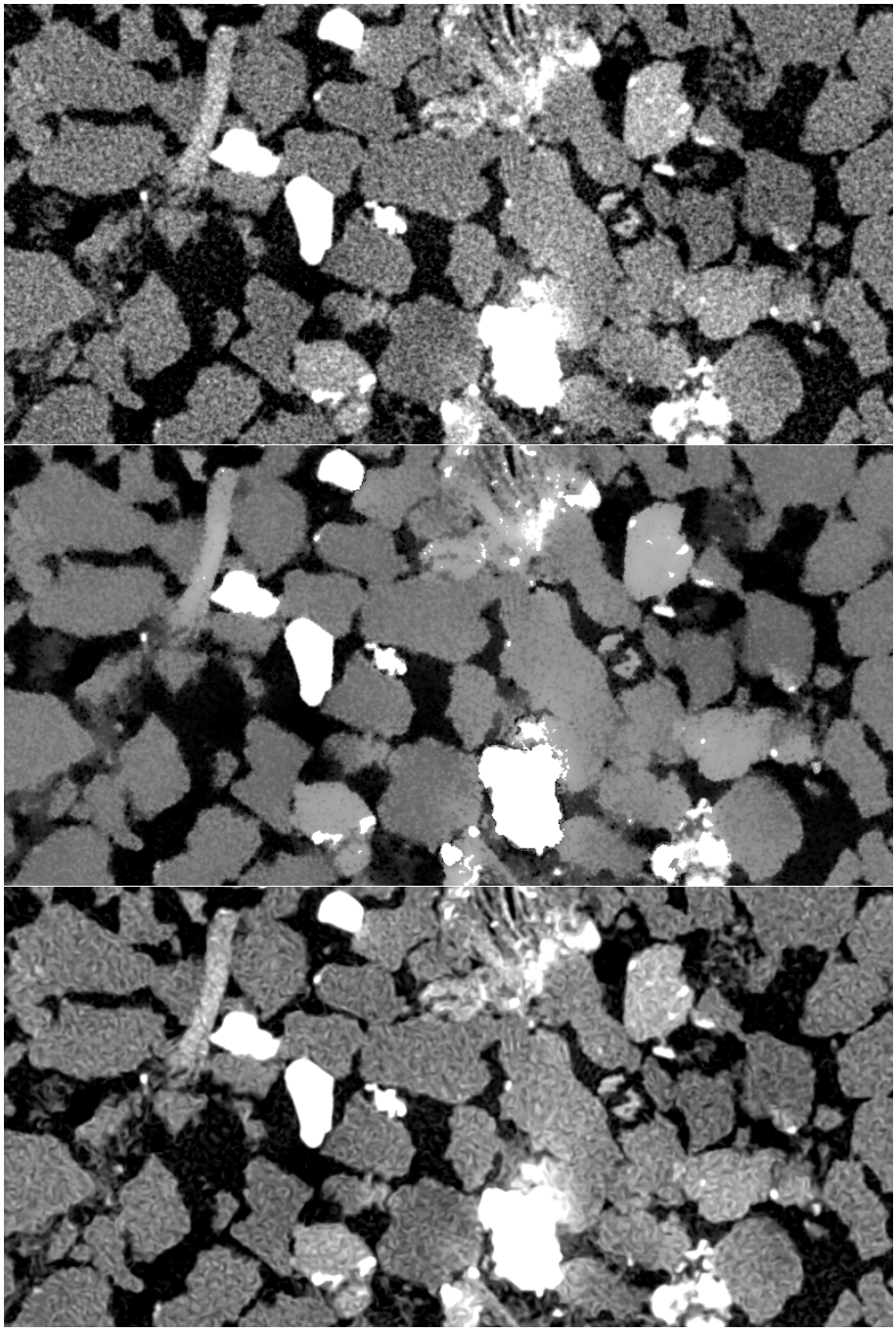

Fig. 5. From top to bottom: 2D slice of the 3D input image on the xy plane (800×400 crop), and outputs of the proposed filter and the anisotropic diffusion algorithm presented in [14].

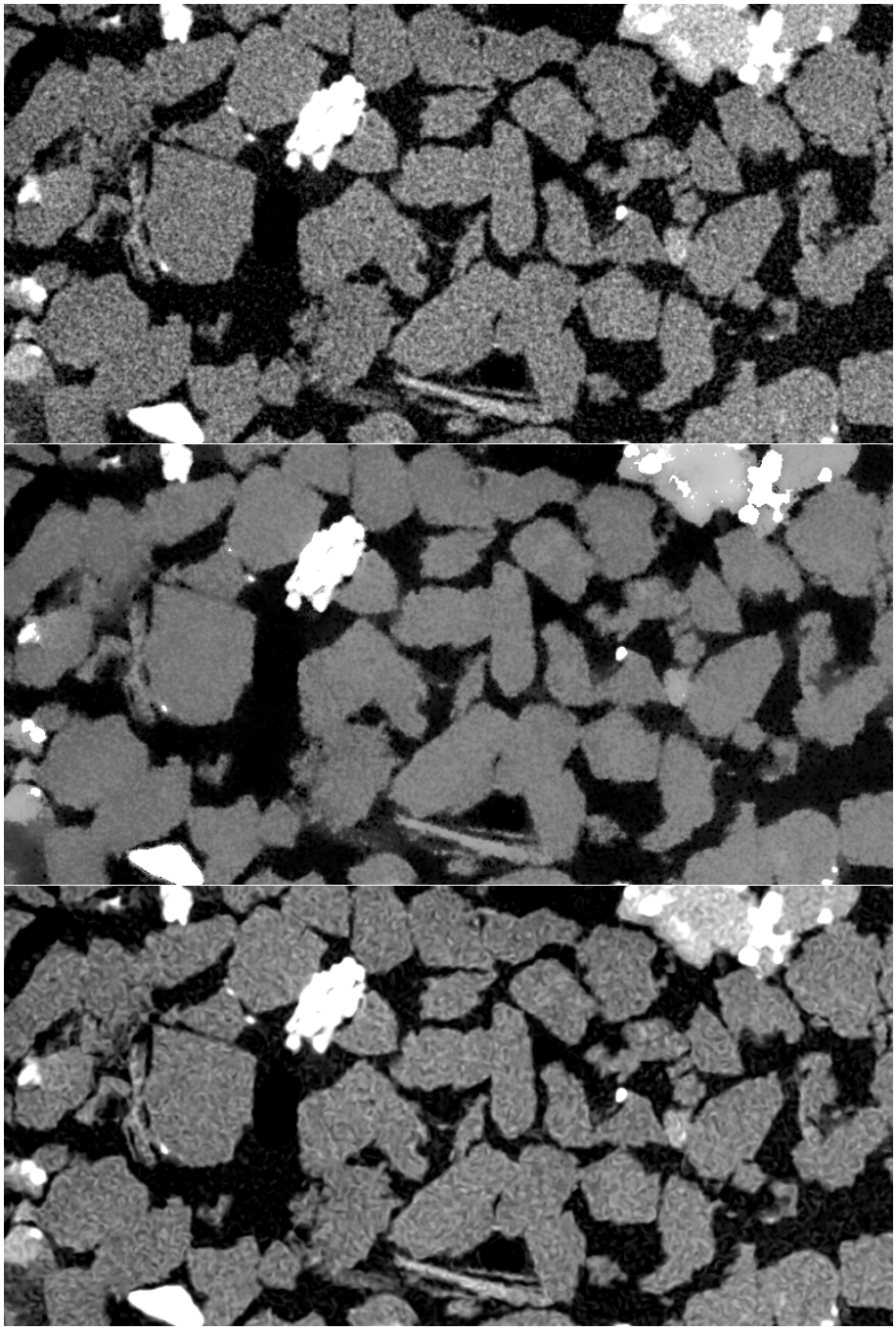

Fig. 6. Same as Fig. 5 for a different input image.

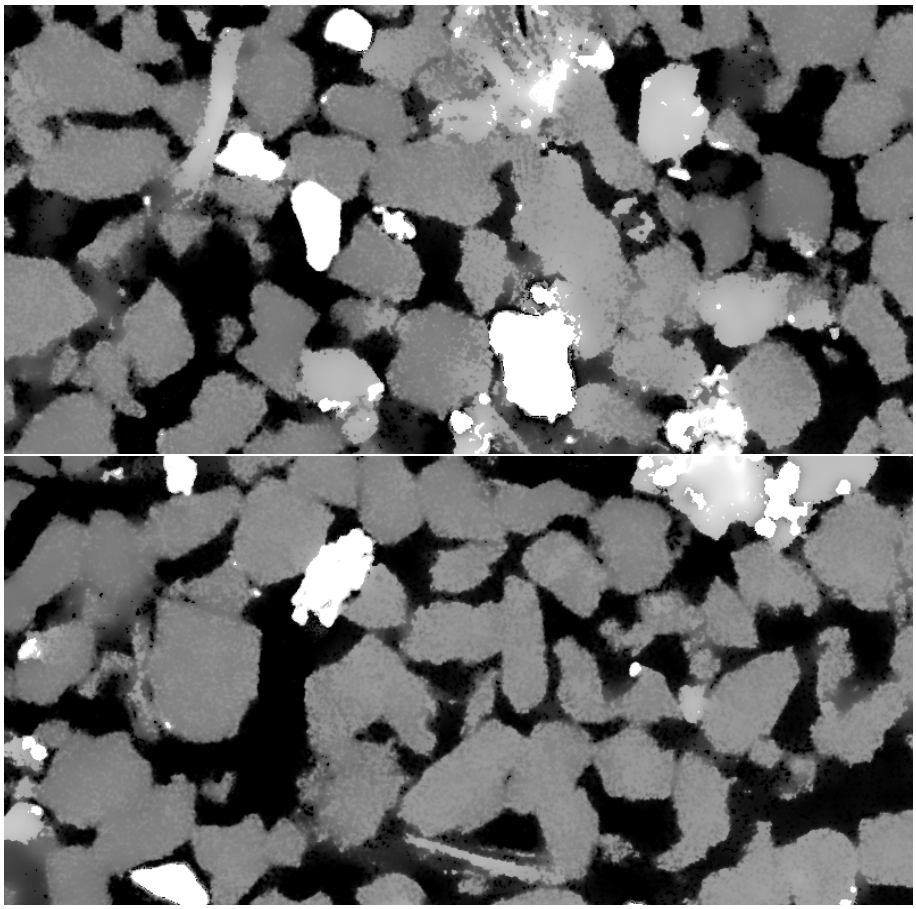

Fig. 7. Output of a double application of the proposed filter for the input images of (top) Fig.5 and (bottom) Fig.6.

diffusion, in terms of amount of noise that can be filtered out while keeping edges sharp, and in terms of computation time.

Acknowledgement. The authors acknowledge FEI Lithicon digital Rock Services for granting permission to publish this paper which is partly funded by RWE Dea and Norwegian Research Council through PETROMAKS project number: 208772/E30, titled: "Physically based three phase relative permeability relations".

References

1. Andrä, H., Combaret, N., Dvorkin, J., Glatt, E., Han, J., Kabel, M., Keehm, Y., Krzikalla, F., Lee, M., Madonna, C., et al.: Digital rock physics benchmarks - part i: Imaging and segmentation. Computers & Geosciences **50** (2013) 25–32
2. Andrä, H., Combaret, N., Dvorkin, J., Glatt, E., Han, J., Kabel, M., Keehm, Y., Krzikalla, F., Lee, M., Madonna, C., et al.: Digital rock physics benchmarks - part ii: Computing effective properties. Computers & Geosciences **50** (2013) 33–43
3. Dvorkin, J., Armbruster, M., Baldwin, C., Fang, Q., Derzhi, N., Gomez, C., Nur, B., Nur, A.: The future of rock physics: computational methods vs. lab testing. First Break **26** (2008)
4. Knackstedt, M.A., Latham, S., Madadi, M., Sheppard, A., Varslot, T., Arns, C.: Digital rock physics: 3d imaging of core material and correlations to acoustic and flow properties. The Leading Edge **28** (2009) 28–33
5. Varslot, T., Kingston, A., Myers, G., Sheppard, A.: High-resolution helical cone-beam micro-ct with theoretically-exact reconstruction from experimental data. Medical physics **38** (2011) 5459–5476
6. Aujol, J.F., Aubert, G., Blanc-Féraud, L., Chambolle, A.: Image decomposition into a bounded variation component and an oscillating component. Journal of Mathematical Imaging and Vision **22** (2005) 71–88
7. Tschumperle, D., Deriche, R.: Vector-valued image regularization with pdes: A common framework for different applications. Pattern Analysis and Machine Intelligence, IEEE Transactions on **27** (2005) 506–517
8. Gilboa, G., Sochen, N., Zeevi, Y.Y.: Estimation of optimal pde-based denoising in the snr sense. Image Processing, IEEE Transactions on **15** (2006) 2269–2280
9. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Computer Vision, 1998. Sixth International Conference on, IEEE (1998) 839–846
10. Kornprobst, P., Tumblin, J.: Bilateral filtering: Theory and applications. Foundations and Trends in Computer Graphics and Vision **4** (2009) 1–73
11. Vliet, J.V., Young, I., Verbeek, P.: Recursive gaussian derivative filters. In: International Conference on Pattern Recognition. Volume 1., IEEE (1998) 509–514
12. Getreuer, P.: A survey of gaussian convolution algorithms. Image Processing On Line **3** (2013) 276–300
13. Michaelis, M., Sommer, G.: A lie group approach to steerable filters. Pattern Recognition Letters **16** (1995) 1165–1174
14. Frangakis, A.S., Hegerl, R.: Noise reduction in electron tomographic reconstructions using nonlinear anisotropic diffusion. Journal of structural biology **135** (2001) 239–250
15. Abràmoff, M.D., Magalhães, P.J., Ram, S.J.: Image processing with imagej. Biophotonics international **11** (2004) 36–43

16. Porikli, F.: Constant time O(1) bilateral filtering. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
17. Chaudhury, K.N., Sage, D., Unser, M.: Fast O(1) bilateral filtering using trigonometric range kernels. Image Processing, IEEE Transactions on **20** (2011) 3376–3382