Proposals on 3D parallel edge-preserving filtration for x-ray tomographic digital images of porous medium core plugs

S S Arsenyev-Obraztsov, E A Volkov and G O Plusch
National University of Oil and Gas, 65 Leninsky Prospekt, Moscow, Russia

Abstract. Precise estimation of the effective petrophysical characteristics for the oil or gas-bearing reservoir plays a vital role in production control problems. It is preferable to use nondestructive measurement methods for porosity, conductivity, geomechanical modulus, and some other parameter fields. For the estimation of anisotropic permeability tensor flows in different directions need to be simulated in the core plug digital image void space, which is very difficult (if possible) to conduct with the real rock sample. The creation of the digital core image includes three stages: construction of its internal structure based on the computed tomography (CT) sinogram, filtration, and segmentation. Routine practice is the filtration of 2D slices pack of the core plug CT image because of a lack of computational power and memory limitations. Unfortunately, this can generate a directional orientation error orthogonal to the pack of slices. The total 3D edge-preserving filtration with the usage of two approaches: modified implicit anisotropic diffusion and discrete orthogonal transforms can reduce this error. Universal code based on the MPI+OpenAcc programming paradigm was tested on different high-performance computing systems incorporating various accelerators like GPGPU and heterogeneous processors such as Sunway 26010.

1. Introduction
Digital analysis of the oilfield core plugs can, to a great extent, replace traditional methods of laboratory examinations and significantly reduce the time required for them. Researchers have the opportunity to understand complex structure of the reservoir better since such an analysis allows correctly take into account the complex geometry of the void space, the morphology of the rock particles, the mineralogical makeup, and based on this information further simulate the flow of fluids at the scale of connected pores.

One of several tools for digital core analysis is CT, which is a method for constructing the internal structure of almost any transparent for some physical waves substance under the assumption that it comprises materials with different radiation absorption coefficients. Tomography is a method of visualization of an object by sectioning. A tomographic survey creates a stack of 2D images of slices of a certain thickness, from which the 3D digital model can be created. Micro-computed tomography (microCT) is a variant of the CT in which the spatial resolution is about a micron. If x-ray radiation is used to build a digital image, then it is called microcomputer x-ray tomography. Recently appeared microCT scanners with a nanoscale resolution which can be of interest for the examination of flow in the low permeability rocks.
For highly fragmented images, it is difficult to construct the boundaries of void space in the porous media. Impulse noises and fine image granularity makes the further flow simulation on the segmented pack of slices numerically very hard because of the time required for the calculations. That is why it is necessary to filter the images of slices before the segmentation. The noise in tomographic images has a diverse nature. It may occur because of the imperfectness of the CT scanner’s matrix, the mechanics of x-ray exposition table motion, the computational round-off errors, and the ill-posedness of the inverse numerical Radon transform.

Most of the smoothing filters are not suitable for processing core plug tomograms, as they displace and flatten out boundaries. They change the morphology of pores, which makes almost meaningless determining effective petrophysical characteristics by numerical simulation of different physical processes on a digital 3D core plug model.

Since CT scan technology origins from medical applications, the common practice is to filter 2D slices in the pack independently [1]. It can be acceptable for large objects in the image, but for a porous medium, such filtering gives directional orientation error (Figure 1) when compounding slices into a 3D digital model. In Fig 1b the directional orientation error, generated by numerical 2D slices pack filtration procedure can be easily seen. Generated disturbances can severely influence the numerical estimation of petrophysical properties such as permeability, geotechnical modules, and some other.

![Figure 1. (a) original image, (b) 2D slice filtering with vertical orientation error, (c) 3D filtering](image)

To increase the quality of the digital image, filtering, segmentation, and numerical simulation can be included in the cycle of inverse problem solution - adapting digital core plug simulation results to the real laboratory tests. RAM capacity required for constructing 3D images can reach
several dozens (hundreds) gigabytes even for an ordinary grayscale image coding. Such problems can be solved only by High-Performance Computing (HPC) clusters. For this purpose, modern heterogeneous HPC systems based on (CPU+GPGPU) or (CPU+accelerators) architecture or even on homogeneous clusters based on heterogeneous CPUs like the Sunway TaihuLight can be used. Because of many different architectures and approaches, the Fortran + MPI + OpenACC programming paradigm was chosen because of its portability and performance. Fortran was selected because of its internal parallelism achieved by including Coarray Fortran (usually based on MPI) in its new standards, which make parallel programming more straightforward.

Two approaches for a 3D edge-preserving filtration: modified implicit anisotropic diffusion and a family of discrete orthogonal transforms were tested.

2. Modified implicit anisotropic diffusion filter
One of the image smoothing algorithms that preserve edges is Perona and Malik anisotropic diffusion filter [2]. It models the diffusion of the voxels (pixels) intensity in a digital image with a specially selected diffusion tensor \( c(r,t) \), which is defined so that the smoothing process does not affect the existing edges. The diffusion equation for our case is:

\[
I(r,t) = \text{div}(c(r,t)\nabla I(r,t)),
\]

\[
I(r,0) = I_0(r), \quad r \in \Omega,
\]

(2.1)

where \( r \) is a point in the spatial domain \( \Omega \), \( I_0(r) \) is the original (initial) image intensity field, and \( I(r,t) \) is an image intensity time derivative.

The diffusion equation is defined in a finite domain \( \Omega \); therefore, boundary conditions on \( \partial \Omega \) are needed. A natural approach is to use boundary conditions conserving image intensity, or exclude boundaries and solve the initial value problem on the \( n \)-dimensional tours as an initial value problem.

Perona and Malik proposed to use the norm of the image intensity gradient \( ||\nabla I|| \) as an estimator for the edge presence at the interior points of \( \Omega \). The higher the gradient, the more possibility of edge presence in the point. So manual construction a diffusion coefficient (tensor) in the form that will not flatten gradient norm is needed:

\[
c(r,t) = g(||\nabla I(r,t)||),
\]

(2.2)

where \( g(x) \) is a decreasing function mapping to the interval \((0,1]\). Perona and Malik proposed two functions:

\[
g(x) = \exp\left(-\frac{(x/\gamma)^2}{2}\right), \quad g(x) = \left(1 + \frac{x}{\gamma}\right)^{-1},
\]

(2.3)

where \( \gamma \) is a filter parameter determined either experimentally or by some heuristic "noise" estimator (for example, the Canny method [3]). Parameter \( \gamma \) acts as a "barrier" for the noise importance level. From (2.3) it can be seen that with a monotonous decreasing \( \gamma \), the measure of the point presence in the edge increases. Therefore, its intensity will not change too much. There are some other forms of this function [4].

2.1. Explicit method
Perona and Malik proposed to solve the quasilinear parabolic equation (2.1) by the explicit form of the finite-difference (FD) method. The scheme was constructed by analogy to the spatial FD discretization for equations with variable coefficients. For example, a part of the FD operator
in the spatial domain for the \( x \) coordinate is:

\[
\frac{\partial}{\partial x} \left( c(x, y, z, t) \frac{\partial I(x, y, z, t)}{\partial x} \right) \approx \frac{1}{(\Delta x)^2} \left( c(x + \frac{\Delta x}{2}, y, z, t) \cdot (I(x + \Delta x, y, z, t) - I(x, y, z, t)) - 
\right. \\
\left. c(x - \frac{\Delta x}{2}, y, z, t) \cdot (I(x, y, z, t) - I(x - \Delta x, y, z, t)) \right).
\]

(2.4)

The spatial discretization step size along each direction is set equal to the distance between pixels so \( \Delta x = \Delta y = \Delta z = h \). For the simplicity let \( h = 1 \). In \([2, 5]\) it was shown that scheme (2.4) will be numerically stable if \( 0 \leq \Delta t \leq 0.25 \).

To diminish the number of calculations, Perona and Malik proposed to use the length of gradient projection on the corresponding axis of the FD grid instead of gradient norm \([2]\). In matrix form, the explicit 3D finite-difference scheme will be:

\[
I_{i,j,k}^{p+1} = I_{i,j,k}^p + \Delta t \left( g(|N_{i,j,k}^p|) N_{i,j,k}^p + g(|S_{i,j,k}^p|) S_{i,j,k}^p + g(|E_{i,j,k}^p|) E_{i,j,k}^p + 
\right. \\
\left. + g(|W_{i,j,k}^p|) W_{i,j,k}^p + g(|D_{i,j,k}^p|) D_{i,j,k}^p + g(|U_{i,j,k}^p|) U_{i,j,k}^p \right),
\]

(2.5)

where \((i, j, k, p)\) are indices of the FD grid for \((x, y, z, t)\), respectively. In this scheme the FD operators \(N_{i,j,k}^p, S_{i,j,k}^p, E_{i,j,k}^p, W_{i,j,k}^p, D_{i,j,k}^p, U_{i,j,k}^p\) are:

\[
N_{i,j,k}^p = I_{i-1,j,k}^p - I_{i,j,k}^p, \quad S_{i,j,k}^p = I_{i+1,j,k}^p - I_{i,j,k}^p, \quad E_{i,j,k}^p = I_{i,j+1,k}^p - I_{i,j,k}^p, \\
W_{i,j,k}^p = I_{i,j-1,k}^p - I_{i,j,k}^p, \quad D_{i,j,k}^p = I_{i,j,k-1}^p - I_{i,j,k}^p, \quad U_{i,j,k}^p = I_{i,j,k+1}^p - I_{i,j,k}^p.
\]

(2.6)

The mnemonic notation \(N, S, E, W, D, U\) mean – North, South, East, West, Down, and Up.

This FD scheme is stable only for rather small time steps. It is unacceptable because CT of one core plug generates a bunch of slices, producing about a thousand of 2D images. The required computer memory depends on the used bit-coding.

2.2. Implicit method

Application of the implicit FD scheme can increase numerical stability:

\[
I_{i,j,k}^p = I_{i,j,k}^{p-1} + \Delta t \left( g(|N_{i,j,k}^p|) N_{i,j,k}^p + g(|S_{i,j,k}^p|) S_{i,j,k}^p + g(|E_{i,j,k}^p|) E_{i,j,k}^p + 
\right. \\
\left. + g(|W_{i,j,k}^p|) W_{i,j,k}^p + g(|D_{i,j,k}^p|) D_{i,j,k}^p + g(|U_{i,j,k}^p|) U_{i,j,k}^p \right).
\]

(2.7)

This discretization leads to a system of nonlinear equations, which can be solved, for example, by the Newton method, and this will increase the execution time. Such an approach will ensure the high accuracy of the PDE numerical solution, but our task is different: it is to prepare the image for the subsequent segmentation. Besides, filtration use double-precision arithmetic, but the result will be converted into the desired image bit-coding with a loss of accuracy because of rounding. Linearization of the system (2.7) is conducted by calculating the diffusion coefficient \(-c\) on the previous time layer. It is natural to suppose for the edge-preserving filter that the
diffusion coefficient should not change too much in time. Therefore the resulting scheme (2.8) will be unconditionally stable [5, 6]:

\[
I_{i,j,k}^{p+1} = I_{i,j,k}^p + \Delta t \left( g(|N_{i,j,k}^p|) N_{i,j,k}^{p+1} + g(|S_{i,j,k}^p|) S_{i,j,k}^{p+1} + g(|E_{i,j,k}^p|) E_{i,j,k}^{p+1} + g(|W_{i,j,k}^p|) W_{i,j,k}^{p+1} + g(|D_{i,j,k}^p|) D_{i,j,k}^{p+1} + g(|U_{i,j,k}^p|) U_{i,j,k}^{p+1} \right)
\] (2.8)

Instead of solving the boundary value problem, if the 3D digital image \((N \times M \times L)\) is thought as a 4D torus surface (2.9) the problem for our PDE will be converted into the initial value one.

\[
I_{i,j,k}^{\text{torus}} = I_i \ (\text{mod} \ N), j \ (\text{mod} \ M), k \ (\text{mod} \ L),
\] (2.9)

In [7] it was shown that the linear operator of the system (2.8) with torus initial condition does not change the total image intensity it is symmetric and positive definite. Therefore, the preconditioned conjugate gradient (PCG) method can solve the system of linear equations (2.8). This approach gives the flexibility to choose the required accuracy in the solution process. It allows to control the termination of iterations and provides an additional acceleration of the filtering procedure. According to the routine method, the image obtained at the previous time step is used as the initial solution guess for the PCG method.

2.3. Filter stopping criteria and parameter estimation process

Usually, the relative residual is used as a stopping criterion for the PCG method. Let \(r^l_p\) be the residual of the iteration \(l\) at the time step \(p\): \(r^l_p = I^p_l - I^{l-1}_p\), where \(I^p_l\) is an image vector (1D representation of 2D/3D image). Then the stopping condition with the threshold value \(\tau\) is:

\[
||r^l_p||/||I^p_l|| < \tau.
\] (2.10)

The non-parametric stopping criterion can be presented in the following form. Iterations will continue until the new value of the residual differs from the zero residual in the selected image bit-encoding unit.

\[
\max_{1 \leq i \leq n} |(r^l_p)_i| = ||r^l_p||_\infty < 1.
\] (2.10)

Numerical experiments show that the application of this criterion significantly reduces the number of iterations required to solve the problem.

This approach is obvious. Pixel brightness (color) is an integer value, so if the infinity norm of the residual vector is less than one bit-encoding digit, then further refinement is not required since the fractional parts will be round-off when converting the result to the image integer presentation. The proposed filter scheme allows us to make arbitrary time steps, but impulse noise may remain in the image due to the large step sizes [7]. There exist some modifications of the Perona and Malik anisotropy functions, which give better results on impulse noise filtering [4]. However, such alterations are much more complicated and resource-consuming for the numerical calculation.

2.4. Filter modification

Two parameters are needed for the creation of an effective filter. The time step \(\Delta t\) can be interpreted as the desired one-step image details blurring or flattering level. The larger the time step, the less processed image minuteness. This means that small specific areas in the image will merge with the large ones containing them or adjacent to them. Parameter \(\gamma\) (scalar or vector) in functions (2.3) can be viewed as a noisiness level. For better results, it is proposed to conduct several filtering iterations with different parameters of the anisotropy function with the constant time step \(\Delta t\).
Let us define the local maximum and minimum differences of intensities for an arbitrary point of the image at the time step \( p \):

\[
\eta_p(r) = \min_{v \in Q(r)} \{|I_p^v - I_p^r|\}, \quad \theta_p(r) = \max_{v \in Q(r)} \{|I_p^v - I_p^r|\}, \quad (2.11)
\]

where \( Q(r) \) is an intensity value for the set of \( N, S, E, W, D, U \) adjacent to this point neighbors.

The criteria for the point to be on edge \( \Gamma \) or belong to the flat region in \( \Omega \) is:

\[
\Gamma_p(r, \gamma) = \frac{g_r(\theta_p(r))}{g_r(\eta_p(r))}, \quad \Omega_p(r, \gamma) = g_r(\eta_p(r)) - g_r(\theta_p(r)). \quad (2.12)
\]

Combination of these criteria into one can be an estimator of the classification ambiguity degree:

\[
\phi_p(r, \gamma) = 2 \min \left( \Gamma_p(r, \gamma), \Omega_p(r, \gamma) \right). \quad (2.13)
\]

It can be shown that \( 0 \leq \phi_p(r, \gamma) \leq 1 \). From inequality \( 0 \leq g_r(x) < 1 \) can be easily deduced that \( \phi_p(r, \gamma) \geq 0 \). Let \( r \) be an arbitrary image point, and \( p \) is a number of filter iterations. Two cases are needed to be considered. First - let \( \Omega_p(r, \gamma) \leq \Gamma_p(r, \gamma) \), then \( \phi_p(r, \gamma) = 2\Omega_p(r, \gamma) \) and \( \Omega_p(r, \gamma) = g_r(\eta_p(r))(1 - \Gamma_p(r, \gamma)) \leq 1 - \Gamma_p(r, \gamma) \). The result of summing these two inequalities is \( \Omega_p(r, \gamma) \leq 1/2 \implies \phi_p(r, \gamma) \leq 1 \). Second - let \( \Gamma_p(r, \gamma) \leq \Omega_p(r, \gamma) \), then \( \phi_p(r, \gamma) = 2\Gamma_p(r, \gamma) \) and \( \Gamma_p(r, \gamma) = 1 - \frac{\Omega_p(r, \gamma)}{g_r(\eta_p(r))} \leq 1 - \Omega_p(r, \gamma) \). Again summing these two inequalities results in \( \Gamma_p(r, \gamma) \leq 1/2 \implies \phi_p(r, \gamma) \leq 1 \). As the point \( r \) was arbitrary therefore \( (\forall r)(\phi_p(r, \gamma) \leq 1) \).

If image filtration is needed with the desired flattering \( \Delta t \) and noisiness \( \gamma_0 \) levels. Then the ambiguity degree of an arbitrary image point will be \( \phi_p(r, \gamma_0) \). One of the possible definitions of the ambiguity degree of the total image after \( p \) iterations with fixed \( \gamma_0 \) is:

\[
\Phi_p(\gamma_0) = \text{median}_r(\phi_p(r, \gamma_0)). \quad (2.14)
\]

Filtration must continue until \( \Phi_{p+1}(\gamma_0) \approx \Phi_p(\gamma_0) \) or \( \Phi_{p+1}(\gamma_0) \) reaches the desired level.

Obviously, in the process of filtration, the noise level in the image will decrease. Therefore, it is necessary to change the parameter \( \gamma \) after each or some number of iterations. The computational experiment showed that the local value of the parameter \( \gamma \) for an arbitrary image point \( r \) for the next time step is better to choose as a root of the equation:

\[
g_{\gamma_{p+1}}(\theta_p(r)) = \phi_p(r, \gamma_0). \quad (2.15)
\]

This choice expresses a natural desire to have a small diffusion on the domain internal boundaries. For the next filtration step, choose \( \gamma_{p+1} \), which corresponds to the necessary blurring of all points in the image: \( \gamma_{p+1} = \text{median}_r(\gamma_{p+1}(r)) \). The median estimator selection is justified by its stability to the existence of some points with a high noise level (outliers).

2.5. Numerical result and parallel implementation

It can be seen in Figure 2 that after applying filtration to the core plug CT image, pores edges become smoother because the filter removed impulse noise from the image. At the same time, the edges became sharper as the filter has flattened pixels intensity values both for pores and for solid-phase areas. Such flattening was achieved by selecting the appropriate filter parameters. The filter simulates the diffusion process in non-edge image areas. The nature of the blur can be considered to be the same type as for the blurring generated by the Gaussian filter [2].

During filtration, the global ambiguity of pixels decreases, which can diminish the computational complexity of the next step in the void space of the core plug generation.
Figure 2. (a) part of the original 2D slice image; (b) filtered image $\Delta t = 5, \gamma = 25$.

Figure 3. Change of the 2D slice image local ambiguity in the process of filter iterations (from left to right); black color means that the point is binary-determined (can be in plateau or in the edge), white color – vague points.

for hydrodynamic simulation – binary segmentation. Figure 3 visualizes the change of local ambiguity for every image pixel. For the plateau pixels (void space and porous medium i.e. solid-state skeleton) ambiguity approaches zero making their segmentation simpler and possibly much cheaper in terms of computational resources while the high ambiguity remains only for the edge pixels. This behavior of the ambiguity degree should be examined for possible usage in edge detection and also for the fast and effective segmentation.

Specialists in the experimental measurement of petrophysical rock parameters estimate filtration results as quite reasonable. However, any such assessment is subjective. For the objective evaluation, it is necessary to compare fluid flow simulation conducted on the digital core plug to the results of permeability and porosity tests executed in a laboratory.

The digital filter code was specially written for HPC usage under the paradigm ”MPI+OpenACC” with implementation in Fortran. For coding, the most common OpenACC pragmas were used for better code portability across different CPU, heterogenic CPU and GPGPU architectures and various OpenACC compatible compilers. The proposed algorithm can be divided into two main sequential blocks: matrix generation for the linear algebraic system and filter parameters estimation, and then the calculation of the approximate solution for this system with the PCG method. Each of these two blocks can be easily paralleled. Independent
Initialization

While $|\Phi^{p+1}(\gamma_0) - \Phi^p(\gamma_0)| > \varepsilon$
Construct matrix for linear system using $\gamma$
Determine $\gamma_{new} := \gamma^{p+1}$
Init CG variables
While $||r|| < 1$
CG step
Loop end
$\gamma := \gamma_{new}$
Loop end
End

Figure 4. Dashed border is used for blocks that paralleled with directives $\texttt{!acc parallel}$ or $\texttt{!acc kernels}$. The solid border is applied for blocks containing comments on the OpenACC usage.

Threads associated with every image pixel can parallel the matrix generation process that gives a tremendous performance potential for GPGPU type processors. Due to the sparseness and symmetry of the matrix wide range of different storage schemes can be used. The flowchart of the parallelization algorithm for the proposed method is represented in Figure 4.

3. Filters based on an integral transformation
The quality and computational efficiency of the segmentation process depends on the level of noise in the digital image and the level of its high granularity. Therefore, it is necessary to pre-filter the CT sinogram – the result of the inverse Radon transform. This process will not only allow to remove the noise of different physical and computational nature but also perform a “preliminary segmentation” or diminish the high granularity of the digital core plug.

The positive feature of filters based on orthogonal transforms is the speed of numerical calculations. It is especially crucial if the raw data exceeds several hundred gigabytes.

This work aims to develop fast parallel algorithms for 3D digital images total filtration rather than filter a sequence (pack) of 2D sections further combined into the 3D image. This approach is more demanding for computer memory. However, it allows to get rid of the vertical orientation effect that occurs if filtering packs of planar slices. It is essential since this effect can lead to a
severe anisotropy of petrophysical parameters fields obtained by computer simulation of various physical processes on 3D digital models of porous medium core plugs.

For initial testing, the direct and inverse discrete Fourier (DFT), and cosine (DCT) transforms, defined on the 3D mesh \((N \times M \times L)\) were used. Direct transforms are:

\[
\hat{I}(\hat{x}, \hat{y}, \hat{z}) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{L-1} I(x, y, z) \exp \left( -2\pi i \left( \frac{x\hat{x}}{N} + \frac{y\hat{y}}{M} + \frac{z\hat{z}}{L} \right) \right),
\]

\[
\hat{I}(\hat{x}, \hat{y}, \hat{z}) = \phi(\hat{x}, N)\phi(\hat{y}, M)\phi(\hat{z}, L) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{L-1} I(x, y, z) \cos \left( A(x, \hat{x}) \right) \cos \left( B(x, \hat{x}) \right) \cos \left( C(x, \hat{x}) \right),
\]

where \(A(x, \hat{x}) = \hat{x}(2x + 1) \frac{\pi}{2N}, \quad B(x, \hat{x}) = \hat{y}(2y + 1) \frac{\pi}{2M}, \quad C(x, \hat{x}) = \hat{z}(2z + 1) \frac{\pi}{2L}.

Inverse transformations for them are:

\[
I(x, y, z) = \frac{1}{MN} \sum_{\hat{x}=0}^{N-1} \sum_{\hat{y}=0}^{M-1} \sum_{\hat{z}=0}^{L-1} \hat{I}(\hat{x}, \hat{y}, \hat{z}) \exp \left( 2\pi i \left( \frac{x\hat{x}}{N} + \frac{y\hat{y}}{M} + \frac{z\hat{z}}{L} \right) \right),
\]

\[
I(x, y, z) = \phi(x, N)\phi(y, M)\phi(z, L) \sum_{\hat{x}=0}^{N-1} \sum_{\hat{y}=0}^{M-1} \sum_{\hat{z}=0}^{L-1} \hat{I}(\hat{x}, \hat{y}, \hat{z}) \cos \left( A(x, \hat{x}) \right) \cos \left( B(x, \hat{x}) \right) \cos \left( C(x, \hat{x}) \right),
\]

where

\[
\phi(\xi, K) = \begin{cases} \frac{1}{\sqrt{K}}, & \xi = 0 \\ \frac{\xi}{\sqrt{K}}, & 1 \leq \xi \leq K - 1 \end{cases}.
\]

DCT can be expressed in terms of DFT [8] as:

\[
\hat{u}_k = Re\{\phi(k, N) \cdot \exp \left( -j\frac{\pi k}{2N} \right) \cdot \mathcal{F}(y_k)\},
\]

where

\[
\begin{align*}
y_n &= u_{2n}, & 0 \leq n \leq \frac{N}{2} - 1 \\
y_{N-1-n} &= u_{2n+1}, & 0 \leq n \leq \frac{N}{2} - 1.
\end{align*}
\]

A distinctive feature of these transformations is the ability to represent a multidimensional Fourier transform as a finite sequence of one-dimensional (here, \(\mathcal{F}_i - \text{DFT}\)):

\[
\hat{u}_{k_0,k_1,k_2} = \mathcal{F}_0(\mathcal{F}_1(\mathcal{F}_2(u_{j_0,j_1,j_2}))).
\]

Illustration of the cosine transform-based filter is presented on a simple model with regular additive noise (Figure 5).

### 3.1. 3D orientation effect

Example of 3D orientation effect, generated by the conventional 2D slice by slice pack filtering is shown in (Figure 6 (a)). Synthetic input data was generated by superimposing regular additive noise on the 3D image created from a pack of identical 2D slices from the previous example. Then conventional 2D slice filtering was performed on a rectangular cuboid. From the resulting 3D array, the vertical section was taken. The presence of regular noise is visible on it.

For suppression of the "ort-directional" residual noise, it is necessary to apply 3D parallel filters created with the usage of some discrete transform. The selection of an integral transform depends on several properties: edge keeping, computational effectiveness, and easiness for...
Figure 5. (a) original image, (b) noise disturbed image, (c) image, filtered by DCT.

Figure 6. vertical sample section: (a) after 2D slice filtering, and (b) after 3D total filtration.

parallelization. Figure 6 (b) presents the result of filtering based on the three-dimensional discrete cosine transform. This filtration severely diminishes regular noise.

For the filters mentioned above, there exist fast transformation algorithms: Fast Fourier Transform (FFT), Fast Discrete Cosine Transform (FDCT). The usage of fast transforms can significantly increase computational efficiency. Besides, they allow rather simple parallelizing. Furthermore, procedures can be called from precompiled libraries adapted for specific supercomputer architectures like xMath library for the SW26010 processor.

3.2. Parallel implementation

The amount of memory storing total 3D image raw data for further processing has significantly increased and needs the usage of high-performance computational systems. Therefore some "universal" approach for conducting calculations on different supercomputer platforms is needed.

One of the most advantageous programming paradigms is "MPI + OpenACC". The latest versions of the OpenACC allows to use it on both multicore and many-core processors. Also, it was adopted for several different computational accelerators. This approach gives the possibility to use the full computational power of modern GPGPUs (Nvidia, AMD), numerical accelerators such as Intel Xeon Phi, and heterogenic processors as Sunway SW26010. This paradigm provides painless portability from one high-performance computational system to another.

The basic idea of parallelizing fast multidimensional cosine transform for the shared/distributed memory systems is based on the following facts: DCT can be calculated by using FFT [8]; multidimensional transform can be executed by superposition of one-dimensional FFT [9]; existence of architecture-dependent parallelized 1D FFT precompiled routines [8].

Let us consider the implementation of this approach on the SW26010 processor, which can serve as a guinea pig for testing different algorithms. First, divide the input data into four parts for each MPE core interconnected by MPI. Second, on MPE cores with CPE matrix calculate FFT along X and Y axes for all subblocks of the 3D image. In switching from X to Y direction, it
is necessary to transpose the array of pixels to speed up the calculation cycles. Third, conduct transform in the Z direction. For this, a new 3D image subdivision is needed, which induce massive amounts of data exchange between individual nodes (MPE cores). This data transfer can be executed by MPI_Alltoall function.

Sunway SW26010 processor memory is hierarchical; each MPE core has access to 8GB of main memory, 32KB L1 data cache, and 256KB L2 data/instruction cache. Each CPE core has 16 KB L1 instruction cache and 64KB Local Device Memory (LDM). Contrary to the cache memory, LDM can be managed explicitly by a programmer. The data movement between the LDM and the main memory is performed through Direct Memory Access (DMA) to guarantee high transfer efficiency.

In the Sunway version swOpenACC, they included several principal extensions that are new clauses, which can be of interest for parallel 3D FDCT calculation. The most important are:

- **local** – clause, to allocate space on the fast LDM memory of CPE thread;
- **swap/swapin/swapout** – clauses, to transpose array along specified index.

The *swap* directive which allows loading arrays transposed relative to a given index directly to the fast LDM memory significantly speeds up 2D blocks FFT computation.

All the above examples were calculated on the SW26010 processor using the previously mentioned approach.

### Conclusion

Executed computer experiments show that conventional slice filtering is not suitable for processing tomographic 3D core plug images because of generated directional orientation error. Moreover, this does not depend on the type of edge-preserving filter. Two 3D filtering approaches have been proposed to improve the quality and speed up subsequent 3D image binary segmentation. For PDE type filtration, the modified implicit anisotropic diffusion filter, filtration stopping criteria, and an algorithm of estimating filter parameters were proposed. Criteria of local and global ambiguity can be used to estimate filtration quality. For the integral transform-based filters a template for further parallelizing them on heterogeneous computer architectures was created. All discussed filters were coded with the paradigm: Fortran + OpenACC + MPI. This approach increases program portability. Prototype programs were tested on cluster nodes equipped with Intel+Nvidia processors and on SW26010 heterogeneous processors.

### Acknowledgments

We want to appreciate the National Supercomputing Center, in Wuxi, China, for giving us the possibility to conduct our calculation on the Sunway TaihuLight supercomputer.

### References

1. Pak T, Archilha N, Mantovani I and Butler I 2019 *Scientific data* 6 ISSN 2052-4463
2. Perona P and Malik J 1990 *IEEE Trans. Pattern Anal. Mach. Intell.* 12 629–639
3. Canny J 1990 *IEEE Trans. on Pattern Analysis and Machine Intelligence* 8(6) 679–698
4. Michel-González E, Cho M H and Lee S Y 2011 vol 10 (BioMed Central) p 47
5. Tauberab C *et al.* 1990 *Applied Numerical Mathematics* 60(11) 1115–1130
6. Rozhdestvenskii B and Yanenko N 1968 *Systems of quasilinear equations* [Sistemy kvazilineinykh uravnenii] (Moscow: Nauka)
7. Arseniev-Obraztsov S S and Plusch G O 2019 *Automation, telemechanization and communication in oil industry* 1(546) 30–40
8. Plonka G, Potts D, Steidl G and Tasche M 2018 *Numerical Fourier Analysis* (Springer)
9. Pratt W K 1991 *Digital Image Processing*, New-York (NY: John Wiley and Sons)