

Bioimage informatics

TomoEED: fast edge-enhancing denoising of tomographic volumes

J. J. Moreno¹, A. Martínez-Sánchez², J. A. Martínez¹, E. M. Garzón^{1,*}
and J. J. Fernández^{3,*}

¹Department of Informatics, University of Almeria, Almeria 04120, Spain, ²Max-Planck Institute for Biochemistry, Martinsried 82152, Germany and ³Spanish National Research Council (CNB-CSIC), Madrid 28049, Spain

*To whom correspondence should be addressed.
Associate Editor: Robert Murphy

Received on March 14, 2018; revised on April 24, 2018; editorial decision on May 18, 2018; accepted on May 22, 2018

Abstract

Summary: TomoEED is an optimized software tool for fast feature-preserving noise filtering of large 3D tomographic volumes on CPUs and GPUs. The tool is based on the anisotropic nonlinear diffusion method. It has been developed with special emphasis in the reduction of the computational demands by using different strategies, from the algorithmic to the high performance computing perspectives. TomoEED manages to filter large volumes in a matter of minutes in standard computers.

Availability and implementation: TomoEED has been developed in C. It is available for Linux platforms at <http://www.cnb.csic.es/%7ejjfernandez/tomoeed>.

Contact: gmartin@ual.es or jj.fernandez@csic.es

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Electron tomography (ET) is an important imaging technique in molecular and cellular biology. ET allows three-dimensional (3D) analysis of the subcellular architecture at the nanometer scale (Lucic *et al.*, 2013). Nevertheless, interpretation of tomographic volumes is often hampered by the typically low signal-to-noise ratio (SNR), especially under cryogenic conditions. Thus, noise reduction is usually applied as a post-processing step (Fernandez, 2012), or even during 3D reconstruction (Chen *et al.*, 2016). Similar filtering needs arise in other 3D electron microscopy techniques for visualization of subcellular organization (Peddie and Collinson, 2014).

Anisotropic non-linear diffusion (AND) is currently the predominant technique in ET owing to its abilities to filter noise with feature preservation (Fernandez and Li, 2003; Frangakis and Hegerl, 2001). It sets the strength and direction of the filtering according to the local structure around each voxel, as estimated by eigen-analysis of the structure tensor:

$$J(I) = \nabla I \cdot \nabla I^T = \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{bmatrix} = \mathbf{VQV}^T \quad (1)$$

with $\nabla I = (I_x, I_y, I_z)$ being the gradient vector of the volume I and \mathbf{VQV}^T denoting the eigen-decomposition of J .

AND follows the diffusion equation, $I_t = \text{div}(\mathbf{D} \cdot \nabla I)$, where I_t denotes the derivative with respect to the time and div is the divergence operator (Supplementary Material). The 3×3 matrix \mathbf{D} is the diffusion tensor and tunes the filtering according to the local structure. \mathbf{D} is built from the eigenvectors \mathbf{v}_i of the structure tensor (Equation 1) and its eigenvalues λ_i (ranking in $[0, 1]$) define the strength of the smoothing along the corresponding direction \mathbf{v}_i .

$$\mathbf{D}(J) = \mathbf{VLV}^T = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3] \cdot \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \cdot [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3]^T \quad (2)$$

For edge preservation, the smoothing along the maximum density variation direction (\mathbf{v}_1) is set as a monotonically decreasing function of the gradient. Typically, $\lambda_1 = 1.0 - \exp(-3.31488/(|\nabla I|/K)^8)$, where the parameter K acts as a gradient threshold that defines edges. By contrast, $\lambda_2 = \lambda_3 = 1$ to highly filter along the two directions with minimum change.

AND is, however, computationally expensive in terms of processing time and memory consumption, which hampers application

to large volumes. Its parallelization is not straightforward due to the dependent stencils involved in the iterative process. Here, we introduce TomoEED, a tool for AND of 3D volumes that has been optimized for execution on standard computers, with reduced memory demands and response time. A GPU version is also included for computers with NVIDIA graphics cards.

2 Implementation

2.1 Fast eigen-analysis of the structure tensors

AND involves massive diagonalization of symmetric 3×3 matrices associated to the eigen-analysis of the structure tensor \mathbf{J} (Equations 1 and 2). This operation is required for all voxels in the volume and as many times as iterations. Standard routines for matrix diagonalization are based on the accurate iterative Jacobi algorithm and are designed mainly for large matrices (Press *et al.*, 2002). Nevertheless, diagonalization of 3×3 matrices can be performed much more efficiently by means of non-iterative analytical calculations, at the expense of limited numerical accuracy (Kopp, 2008). TomoEED makes use of direct analytical calculation of the eigensystems to reduce processing time without practical influence in the denoised results. Further details are in Supplementary Material.

2.2 High performance computing in TomoEED

AND is a memory-bound application. Typical memory requirements in standard implementations amount to eight copies of the volume. This is to hold the input/output volumes and the six components of the symmetric tensor \mathbf{J} (Equation 1), which are also shared (overridden) by \mathbf{D} (Equation 2)).

TomoEED implements an efficient scheme where only one copy of the volume is held in memory and it is gradually updated by Z-planes during the iterative process. An auxiliary sliding window is used to maintain the data needed for the calculation of current Z plane: neighbouring Z-planes and their tensors (Supplementary Fig. S1). This optimized implementation allows making the most of the memory hierarchy and enables denoising of huge datasets in computers with modest amounts of memory.

To exploit the power of modern multicore computers, TomoEED runs in multithreaded mode. Here, the calculation of the current Z-plane is distributed among the threads running in parallel. Each thread processes one subset of Y-rows (Supplementary Figs S1 and S2), which involves the calculation of \mathbf{J} and \mathbf{D} followed by the iteration of the diffusion equation, with thread synchronizations in-between, for all Y-rows in their subset.

TomoEED is well suited to GPU processing as each voxel can be processed independently (with synchronization points between iterations). A CUDA-based implementation is included that maps each voxel to a GPU thread for a massively parallel execution. Additionally, it restructures the layout of \mathbf{J} and \mathbf{D} to increase memory performance on these architectures.

2.3 Automated parameter tuning

The main parameter in AND, K , acts as a threshold on the gradient. Voxels with higher gradient are considered edges to be preserved, thereby decreasing the filtering along the first eigen-direction. This parameter is dataset-dependent, its tuning is not trivial and it is usually set by trial-and-error. TomoEED adopts strategies for its automated, time-varying setup based on the average gradient of the whole 3D volume or a noise subregion. They facilitate user

Table 1. Processing time (s), speedup factors and memory consumption (GB)

Dataset size	Jacobi	Analytic		Speedup			Memory consum.	
		1T	16T	GPU	1T	16T		GPU
256	60.50	33.96	3.78	1.83	1.78	15.99	33.05	0.07
384	205.00	113.89	10.77	4.72	1.80	19.04	43.36	0.23
460	356.33	199.00	18.77	7.65	1.79	18.98	46.57	0.39
512	498.46	275.61	24.18	9.79	1.81	20.61	50.90	0.53
640	961.34	530.20	45.23	19.11	1.81	21.65	50.31	1.02

operation by providing acceptable denoised solutions from which manual refinement can follow (Fernandez *et al.*, 2007).

3 Illustrative results

To illustrate the performance of TomoEED, we have applied it to datasets from different volume electron microscopy disciplines where noise filtering is needed (Supplementary Material). Significant noise reduction and preservation of the main structural biological features are observed.

We have also analyzed the processing time, scalability and memory consumption with datasets of different cubic sizes (256, 384, 460, 512 and 640) on a computer with two octo-core processors Intel Xeon E5-2650 v2 and a NVIDIA GPU Tesla K80. Supplementary Tables S1 and S2 presents a full report of the results. Table 1 summarizes the results. The processing times from 10 iterations of AND obtained with TomoEED using analytical matrix diagonalization with 1 thread (1T), 16 threads (16T) and on the GPU are presented. For comparison, the results using the Jacobi algorithm optimized for 3×3 matrices with 1 thread are included. It can be observed that the analytical diagonalization accelerates the computation in a factor around $1.8\times$ with respect to the Jacobi algorithm. The multithreaded execution on the 16-core machine further reduces the processing time and achieves a final speedup factor in the range $16\text{--}21\times$, with higher values for larger volumes. This translates into computing times much lower than a minute for all datasets. The GPU version achieves outstanding speedup factors ($33\text{--}50\times$), with times lower than 20s. For comparison with standard programs, we applied AND within IMOD (Kremer *et al.*, 1996) and demonstrated that TomoEED is much faster, especially with analytical diagonalization, and requires $8\times$ less memory (Supplementary Table S3).

The limited accuracy of the analytical diagonalization does not produce noticeable visual differences in the denoised solutions. For quantitative assessment, the relative error between the solutions obtained with the accurate Jacobi algorithm and the analytical strategy was computed, and it turned out to be negligible (Supplementary Table S4). Moreover, SNR and sharpness of the denoised solutions confirmed that there are no practical differences between the two diagonalization strategies (Supplementary Table S5).

4 Conclusion

TomoEED is a powerful and efficient software tool for fast feature-preserving noise reduction in different volume electron microscopy disciplines. It is based upon anisotropic non-linear diffusion. Its mechanisms for automated parameter setup simplify user operation. Its optimized implementation enables its application to large datasets on standard computers, with reduced turnaround times and memory demands.

Funding

Grants TIN2015-66680 and SAF2017-84565-R (AEI/FEDER, UE) and Fundación Ramón Areces.

Conflict of Interest: none declared.

References

- Chen, Y. et al. (2016) FIRT: filtered iterative reconstruction technique with information restoration. *J. Struct. Biol.*, **195**, 49–61.
- Fernandez, J.J. (2012) Computational methods for electron tomography. *Micron*, **43**, 1010–1030.
- Fernandez, J.J. and Li, S. (2003) An improved algorithm for anisotropic diffusion for denoising tomograms. *J. Struct. Biol.*, **144**, 152–161.
- Fernandez, J.J. et al. (2007) Three-dimensional anisotropic noise reduction with automated parameter tuning. *Lect. Notes Comp. Sci.*, **4788**, 60–69.
- Frangakis, A.S. and Hegerl, R. (2001) Noise reduction in electron tomographic reconstructions using nonlinear anisotropic diffusion. *J. Struct. Biol.*, **135**, 239–250.
- Kopp, J. (2008) Efficient numerical diagonalization of hermitian 3×3 matrices. *Int. J. Mod. Phys. C*, **19**, 523–548.
- Kremer, J. et al. (1996) Computer visualization of three-dimensional image data using IMOD. *J. Struct. Biol.*, **116**, 71–76.
- Lucic, V. et al. (2013) Cryo-electron tomography: the challenge of doing structural biology in situ. *J. Cell Biol.*, **202**, 407–419.
- Peddie, C.J. and Collinson, L.M. (2014) Exploring the third dimension: volume electron microscopy comes of age. *Micron*, **61**, 9–19.
- Press, W.H. et al. (2002). Numerical recipes in C. In: *The Art of Scientific Computing*, 2nd edn. Cambridge University Press, Cambridge.