TomoEED: Fast Edge-Enhancing Denoising of Tomographic Volumes

J. J. Moreno ¹, A. Martinez-Sanchez ², J.A. Martinez ¹, E.M. Garzon ¹, J. J. Fernandez ³

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

March 2018

Contact: jjfernandez.software@gmail.com

Web site

The package TomoEED contains optimized programs for denoising tomographic volumes with the anisotropic nonlinear diffusion (AND) method, using the EED (Edge-Enhancing Diffusion) mode. This mode manages to reduce noise in the volume with good abilities to preserve and enhance the edges. At each voxel of the tomogram the method is capable of detecting the direction of the edges. The method thus applies smoothing along (not across) the edge, hence preserving and enhancing it. For homogeneous areas, the method applies strong Gaussian-like smoothing.

The package has been optimized to reduce memory requirements (with a reduction factor of about $8 \times$ compared to standard implementations). Moreover, multithreading and code optimization techniques are used to take advantage of current multicore computers and reduce the processing time. Finally, options to assist the users to tune the parameters are available. An ultrafast GPU version is also provided in the package.

Table of Contents

1. Installation
2. Description
3. Usage
4. Options and parameters
5. Examples
6. Advices and comments
7. References
1 Installation

• Uncompress the file TomoEED.zip.

• You will find these directories:
  –  `bin`, with the Linux x86_64 executables for CPU and GPUs: `tomoed`, `tomoed_gpu` and an OSX (compiled under OSX 10.6.8) executable for CPU: `tomoed.osx`
  –  `doc`, with this documentation (`tomoed.pdf`).

• Set up your `PATH` environment variable to have direct access to the executables.
2 Description

The method for denoising tomograms implemented in TomoEED is Anisotropic Nonlinear Diffusion (AND), Edge Enhancing Diffusion (EED) mode. This mode allows significant noise reduction with edge preservation and enhancement.

The program works as follows:

- The program proceeds iteratively and at each iteration it analyzes each voxel based on the gradient magnitude and the local structure. If the gradient magnitude is lower than a given threshold (the parameter $K$ -also known as lambda- of the method), the program applies a strong isotropic Gaussian-like filtering at the voxel. Otherwise, the smoothing along the direction of the maximum density variation (i.e. perpendicular to the edge) is avoided or attenuated. This ensures preservation and enhancement of the edges. The function that defines the actual smoothing in the direction perpendicular to the edge is the following, which was introduced by Weickert in his pioneering work on AND.

\[ 1 - \exp\left(-3.31488/((\text{gradient}/K)^8)\right) \]

- The method is strongly sensitive to the parameter $K$ (lambda). However, setting it up is not trivial. TomoEED is equipped with options to assist the user to tune this parameter. There are several options:

  - By default, at each iteration the program sets the parameter $K$ as the average gradient found in the whole tomogram. Therefore, the program uses a time-varying value for $K$ (lambda), which is usually large at the beginning and decreases with the iterations as the tomogram is being denoised (Fernandez et al. LNCS’07).

  - With the option `-N`, the parameter $K$ (lambda) is also time-varying and is set as the average gradient found in an area containing only noise (background). This area contains $a \times a \times a$ voxels and is specified by the user introducing the centre of the area $(x, y, z)$ and the box size $a$. Again, $K$ (lambda) turns out to be usually large at the beginning of the process and decreases with the iterations as the tomogram is being denoised. The difference with respect to the previous option is that $K$ (lambda) is tuned with a better ‘reference’: an area containing only noise (background). This, in principle, should provide better estimates of the parameter.

  - With the option `-n`, the parameter $K$ (lambda) is set as the average gradient found in an area containing only noise (background) of the original tomogram. That is, the parameter is not time-varying but remains fixed during all denoising cycles. The noise area contains $a \times a \times a$ voxels and is specified by the user introducing the centre of the area $(x, y, z)$ and the box size $a$.

  - With the option `-k`, the users sets the parameter $K$ (lambda) to a specific value that remains constant for the entire denoising process.

- At the beginning of the process, the tomogram is subjected to a standard Gaussian filtering. This is intended to compute better gradient estimates. By default, the program always applies this filtering, which should be slight (normally with standard deviation in the range $[0.5,1]$; by default 0.5 is taken). It is strongly recommended that this filtering be applied.
This version of the program has been optimized in terms of memory requirements. The memory consumption has been reduced by a factor about $8 \times$ with regard to the standard implementations, which makes the program suited for processing large tomograms or use computers equipped with relatively modest amount of memory. The program only requires memory to keep the input tomogram in float mode, and is updated in-place as the denoising proceeds. Some other structures are used as well, but they require a rather marginal amount of memory. All computations are carried out on-the-fly, thus minimizing the need for storage.

This version of the program has been developed using code optimization and multithreading techniques in order to reduce the processing time.

The GPU version (tomoeed_gpu) exploits the NVIDIA graphics cards in computers where they are available.

A detailed description of the procedures implemented in the program can be found in the following article.

**TomoEED: Fast Edge-Enhancing Denoising of Tomographic Volumes**
JJ Moreno, A Martinez-Sanchez, JA Martinez, EM Garzon, JJ Fernandez.
https://doi.org/10.1093/bioinformatics/bty435

Please, cite this article if you use tomoeed in your works.

Other articles related to the program are the following:

*An improved algorithm for anisotropic nonlinear diffusion for denoising cryotomograms.*
JJ Fernandez and S Li
http://dx.doi.org/10.1016/j.jsb.2003.09.010

*Anisotropic nonlinear filtering of cellular structures in cryoelectron tomography.*
JJ Fernandez and S Li
http://dx.doi.org/10.1109/MCSE.2005.89

*Three-dimensional anisotropic noise reduction with automated parameter tuning. Application to electron cryotomography.*
JJ Fernandez, S Li and V Lucic
http://dx.doi.org/10.1007/978-3-540-75271-4_7
## 3 Usage

The program allows noise reduction based on anisotropic nonlinear diffusion, EED mode, running efficiently on standard multicore computers and GPUs. The program works with a command line user interface that adheres to typical Unix-style practices. To get the list of options (note that some options are not available, or are still pending, in the GPU version – H,N,t –), just type the program name followed by `-h`:

```
tomoeed -h
```

It will show the following information on console:

```
------------------------------------------------------------
Usage: tomoeed [options] input.tomogram output.tomogram
Where:

input/output tomograms: MRC format (modes 0, 1, 2).
  Default: output uses the same mode as input

Optional parameters:

-c Indicates that the input tomogram comes from a previous run
  of the program. Equivalent to use `-g 0'
-f The output tomogram is saved in FLOAT mode, regardless of the
  input mode.
-g sigma Sigma of the initial Gaussian filtering (default: 0.5)
-h Show this help.
-H Use Hyper-Threading if available
-i iter No. Iterations of Anisotropic Nonlinear Diffusion
  (default: 10)
-j Use Jacobi method for structure tensor eigen-analysis.
  (default: fast analytic calculation is used.)
-k lambda Parameter K (lambda). (default: time-varying K calculated
  from average gradient in tomogram.)
-n x,y,z,a Coordinates of a noise area, defined by the centre (x,y,z)
  and the box size (a) to obtain the value for K.
-N x,y,z,a Coordinates of a noise area, defined by the centre (x,y,z)
  and the box size (a) to obtain the time-varying value for K.
-s step Time step. (default: 0.1; maximum value: 0.15).
-t threads Use multithreading (default: as many threads as cores).
-v level Verbosity level: 0 (silent), 1 (low, default), 2 (high).
```
4 Options and Parameters

Mandatory parameters

input/output tomograms
The format of the input tomogram is MRC, any mode (0 - unsigned char; 1 - short; 2 - float), as typically used with standard packages like IMOD. The format of the output tomogram is MRC using, by default, the same mode as the input tomogram. However, forcing the writing of the tomogram in mode 2 (float) is possible with the option ‘-f’, as described below.

Optional parameters for the general user

-c
This option allows the user to indicate that the input tomogram comes from a previous run of the program. If so, the program automatically switches off any initial filtering to avoid any potential blurring of the edges. So this option is equivalent to use ‘-g 0’.

-f
This option forces the program to write the output tomogram in mode 2 (float). By default, the output tomogram is written using the same mode as the input tomogram.

-g <sigma>
This option allows the user to enter the standard deviation for an initial Gaussian filtering of the input tomogram. This initial filtering is intended to remove the fine-grain noise in order to regularize the computation of the gradients and make them less sensitive to noise. A value in the range [0.5,1.0] is advisable. Higher values may blur the structures, which would thus spoil the effects of the subsequent AND process. By default a value of 0.5 is assumed. If no initial Gaussian filtering is wanted, it must be switched off with ‘-g 0’.

-h
The program shows the help information on console and exits.

-H
The program automatically detects whether the cores have hyperthreading technology. However, by default the program does not exploit this feature. This option allows activation of the use of hyperthreading. In that case, the program creates additional threads to a total of twice the number of cores. However, bear in mind that in an application like this all the threads execute the same operations using the same functional units within the core (hence competing for them). As a result, the speedup achieved by the hyperthreading might not be as high as that obtained by two real cores working in parallel. This flag is ignored if the option ‘-t’ is used. This option is not available in the GPU version.
-i <iter>
Number of diffusion iterations. A number of iterations around [10,60] are OK, though it strongly depends on the parameter K (lambda). If K is set to time-varying average gradient in the whole tomogram (i.e. the default behaviour), the number of iterations should be relatively low (10–20). The more iterations the stronger the smoothing. If too many iterations are used, some interesting structural features may turn out to be blurred. By default, 10 iterations are assumed. This parameter is strongly related to the time step (see option ’-s’).

-j
This option makes the program use the iterative Jacobi method for the matrix diagonalization involved in the eigen-analysis of the structure tensor. This method is more accurate at the expenses of processing time. By default, TomoEED performs this operation with fast analytic calculation (about 2× faster), with negligible impact in the resulting denoised volumes.

-k <lambda>
This option allows the user to specify for the parameter K (lambda) of the Edge Enhancing Anisotropic Nonlinear Diffusion (EED). This value will be used throughout the denoising process. That is, it remains constant for all iterations. By default, the program uses a time-varying K set as the average gradient computed from the whole tomogram. The options -k, -n and -N are mutually exclusive.

-n <x,y,z,a>
This option allows the user to specify the coordinates and extension of an area in the tomogram containing only noise (background). The parameter K (lambda) will be set to the average gradient calculated from that background area, and will remain fixed throughout the denoising process. The values entered by the user are the coordinates of the centre of the area (x, y, z) and the box size a. For the coordinates, it is assumed that the first voxel in the tomogram is (1, 1, 1). The options -k, -n and -N are mutually exclusive.

-N <x,y,z,a>
This option allows the user to specify the coordinates and extension of an area in the tomogram containing only noise (background). The parameter K (lambda) will be set to the average gradient calculated from that background area, and will be updated at each iteration from the denoised tomogram. The values entered by the user are the coordinates of the centre of the area (x, y, z) and the box size a. For the coordinates, it is assumed that the first voxel in the tomogram is (1, 1, 1). The options -k, -n and -N are mutually exclusive. This option is not yet available in the GPU version.

-s <step>
This option allows the user to change the time step. The larger the time step, the lower the number of iterations needed. The default value is 0.1, which is the standard value. However, the maximum value for the sake of numerical stability that still works for explicit discretization of partial differential equations is 0.15. So, the user might want to set this parameter to 0.15, and decrease the number of iterations accordingly.
-t `<threads>`

By default, the program automatically detects the number of cores available in the computer and creates as many threads as cores, which will collaborate in the denoising process. In other words, by default the program intends to fully exploit the parallel capabilities of multicore (dual, quad, hexa, etc) computers. This option allows the user to specify a particular number of threads instead of using the default value. In that case, the user should bear in mind that the maximum number of threads should not be higher than the number of cores in the computer, so for a computer with a quad-core, 4 should be the maximum number of threads; similarly, 2 threads should be the maximum for a dual-core (i.e. Intel Core 2 duo) and 8 threads for a computer with 8 cores, etc. If the cores have hyperthreading technology available, additional threads could be used (see the notes for the flag -H). This option is not available in the GPU version.

-v `<verbosity_level>`

Sets the verbosity level: 0 (no verbose), 1 (low, default level), 2 (high). The verbosity level 2 shows the statistics of the denoised tomogram at every iteration. If the parameter K (lambda) is set time-varying, the actual value at each iteration is shown in verbosity levels 1 and 2.

**Summary: The most important parameters**

The most important options for denoising are related to the initial Gaussian filtering (-g), to the parameter K (-k, -n, -N) and the number of iterations (-i). The latter is also related to the time step (-s).
5 Examples

5.1 Examples of commands

1. Denoising using the default initial Gaussian filtering (standard deviation of 0.5) followed by the default 10 iterations of AND with a time step of 0.1. The parameter K gets time-varying values computed from the average gradient of the whole tomogram as it is being denoised. The program works at full computational speed, i.e. internally it creates as many threads as cores available in the computer, which work in parallel collaborating in the filtering. The verbosity level 1 is used, so the program shows the progress of the denoising.

   ```tomoeed inputtomogram.mrc outputtomogram.mrc```

2. Denoising using 40 iterations, using a parameter K set by the user to 2.0. The value of K will remain constant throughout the denoising process. The other parameters are set to their default values.

   ```tomoeed -k 2.0 -i 40 inputtomogram.mrc outputtomogram.mrc```

3. Denoising using 30 iterations, using a parameter K that is automatically set as the average gradient in an area containing only background. This area has $25 \times 25 \times 25$ voxels and is centred at the voxel (120,45,35) of the tomogram (these coordinates could be obtained using a visualization program, such as 3dmod in IMOD). The value of K will remain constant throughout the denoising process. The other parameters are set to their default values.

   ```tomoeed -i 30 -n 120,45,35,25 inputtomogram.mrc outputtomogram.mrc```

4. Denoising using 15 iterations, using a time-varying value for the parameter K computed from an area containing only background. This area has $20 \times 20 \times 20$ voxels and is centred at the coordinates (100,50,75) of the tomogram (these coordinates could be obtained using a visualization program, such as 3dmod in IMOD).

   ```tomoeed -i 15 -N 100,50,75,20 input.mrc output.mrc```

5. The same denoising as above is run using the maximum time step (0.15), adjusting the number of iterations accordingly (10). Note that the diffusion time in this example would be $10 \times 0.15 = 1.5$. This is the same as in the previous example: $15 \times 0.1 = 1.5$. Therefore, a result similar to the previous command is obtained, but with smaller processing time.

   ```tomoeed -i 10 -s 0.15 -N 100,50,75,20 input.mrc output.mrc```

6. Denoising with 10 iterations of EED and all the parameters at default values.

   ```tomoeed inputdata.mrc outputdata_eed10.mrc```

   If the user feels that more iterations are necessary, it is possible to resume (-c) the filtering with additional iterations, say 30 more.

   ```tomoeed -c -i 30 outputdata_eed10.mrc outputdata_eed40.mrc```
7. Denoising using an initial Gaussian filtering with standard deviation of 1.0 followed by 20 iterations of EED using $K = 1.5$. The verbosity level 2 is set up to obtain the statistics at every iteration.

```
tomoeed -v 2 -g 1 -i 20 -k 1.5 inputtomogram.mrc outtomogram20.mrc
```

Continuation of the previous denoising with 30 further iterations, to yield a final denoised tomogram after a total of 50 iterations of EED. In this case, the verbosity level is set up to 0 in order to work silently.

```
tomoeed -v 0 -c -i 30 outtomogram20.mrc outtomogram50.mrc
```

8. Let’s assume that the denoising process is to be run on a computer based on a quad-core processor with hyperthreading technology available. By default, tomoeed will create four threads working in parallel, but it would not take advantage of hyperthreading. To exploit this technology, the flag -H has to be used, which would make the program create eight threads working concurrently. This is an example using the default parameters for the denoising:

```
tomoeed -H inputtomogram.mrc outputtomogram.mrc
```

9. Let us assume the previous command is to be run on a computer with 4 cores, but we have another program running in the computer that is using one core. It may be helpful to set the number of parallel threads to 3, i.e. the three cores that are free. This can be specified with the option ‘-t’. In this case, no hyperthreading is used.

```
tomoeed -t 3 inputtomogram.mrc outputtomogram.mrc
```
5.2 Example of the output report

If verbosity level 2 is selected, the program echoes the options specified in command line and then the following output report is shown on console during the execution:

Initial
Statistics (avg, std, min, max): 1.361 42.951 -1314.725 3854.679

iteration number: 1 / 10
diffusion time: 0.1000
K (lambda): 41.46
Statistics (avg, std, min, max): 1.361 36.797 -1207.343 3596.301

iteration number: 2 / 10
diffusion time: 0.2000
K (lambda): 31.90
Statistics (avg, std, min, max): 1.361 33.204 -1116.880 3365.589

. . .

iteration number: 9 / 10
diffusion time: 0.900
K (lambda): 13.28

iteration number: 10 / 10
diffusion time: 1.0000
K (lambda): 12.34
Statistics (avg, std, min, max): 1.362 24.118 -931.188 2378.515

This report shows on console the progress of the program. First, it shows the statistics (average, standard deviation, minimum, maximum) of the initial volume. Then at every iteration it shows the corresponding diffusion time (i.e. iteration × the time step) and the current statistics. The value of the parameter K is also shown if time-varying K is being used. The average density value in the tomogram should keep approximately constant with the iterations whereas the standard deviation should be progressively reduced.

If verbosity level is set to 1 (the default value), only the index of the current iteration and K (if time-varying) is shown.
6 Advices and comments

- It is advisable to use the initial Gaussian filtering, at least with the default values (standard deviation 0.5), which removes the shot noise and allows more reliable computations of the gradient. A value in the range [0.5, 1.0] works fine. Higher values might yield good results as well, but under the risk of blurring.

- If the parameter K is set to time-varying average gradient in the whole tomogram (i.e. the default behaviour), the number of iterations should be relatively low (10–20) to avoid excessive blurring of the features of interest. As the whole tomogram is considered, background and edges are included in the computation of the average gradient. As a result, the value of K is over-estimated. Reducing the number of iterations then helps to avoid over-denoising.

- Probably, the best way to set the parameter K is using a time-varying value computed from a background area (option -N).

- Memory requirements. Given an input tomogram of $N_x \times N_y \times N_z$ voxels, the program requires a memory space of $4 \times N_x \times N_y \times (N_z + 22)$ Bytes.

Acknowledgements

This work has been possible thanks to grants from the Spanish AEI/FEDER, UE, and Fundacion Ramon Areces.
7 References

- of special interest.

Reference describing the implementation in TomoEED


References on denoising in volumetric electron microscopy with anisotropic/isotropic nonlinear diffusion


References on denoising in volumetric electron microscopy with other methods


References on denoising with nonlinear diffusion


