

# **TOMO3D 2.0: Fast tomographic reconstruction on multicore computers**

J.I. Agulleiro <sup>1</sup>, J.J. Fernandez <sup>2</sup>

<sup>1</sup> Associated unit CSIC+UAL. University of Almeria (UAL). 04120 Almeria. Spain.

<sup>2</sup> National Centre for Biotechnology. National Research Council (CSIC). 28049 Madrid. Spain.

January 2015

Contact: [jjfernandez.software @ gmail.com](mailto:jjfernandez.software@gmail.com)

[Web site](#)

---

Tomo3D implements a multithreaded vectorized approach to tomographic reconstruction that takes full advantage of the computing power in modern multicore computers. It makes use of Pthreads to exploit the multiple cores and vector instructions (SSE or AVX) to benefit from the vector functional units available in them. Tomo3D also overlaps input/output disk operations with computation in order to reduce their latencies over the effective program execution time. Full resolution tomograms are thus generated at high speed on standard computers with no special system requirements. Tomo3D has the most common reconstruction methods implemented, namely WBP and SIRT. It proves to be competitive with current GPU solutions in terms of processing time, in the order of a few seconds with WBP or minutes with SIRT. There are binaries of Tomo3D for Linux, OSX and Windows. Tomo3D reads input tilt-series and writes output tomograms in MRC format. The program is compatible with standard packages, which easily allows integration in the electron tomography workflow.

Tomo3D 2.0 is the version capable of exploiting AVX vector instructions, if present in the processor. Tomo3D 2.0 is only available for Linux at the moment. For OSX and Windows, the previous version – Tomo3D 1.3.4 (released in April 2012) –, which benefits from SSE vector instructions, is provided.

## Table of Contents

1. Installation
2. Description
3. Usage
4. Options and parameters
5. Examples
6. Advices and comments
7. Tomo3D under Windows
8. History
9. References

# 1 Installation

- Uncompress the file `tomo3d.zip`.
- You will find these directories:
  - `bin`, which contains the executable binaries:
    - \* `tomo3d`  
Tomo3D 2.0, 64-bit executable program for Linux. Minimum processor requirements: a 64-bit Intel Pentium 4 or an AMD K8.
    - \* `tomo3d.ia32`  
Tomo3D 2.0, 32-bit executable program for Linux. Minimum processor requirements: an Intel Pentium or an equivalent AMD processor.
    - \* `tomo3d.osx`  
Tomo3D 1.3.4 (April 2012), 64-bit executable program for Intel-based Macs. It was built under OSX 10.6.8 (Snow Leopard).
    - \* `tomo3d.w64`  
Tomo3D 1.3.4 (April 2012), 64-bit executable program for Windows (XP and onwards)<sup>1</sup>. Minimum processor requirements: a 64-bit Intel Pentium 4 or an AMD K8.
    - \* `tomo3d.w32`  
Tomo3D 1.3.4 (April 2012), 32-bit executable program for Windows (XP and onwards)<sup>1</sup>. Minimum processor requirements: an Intel Pentium or an equivalent AMD processor.
  - `doc`, with this documentation.
- Set up your `PATH` environment variable to have direct access to the program(s).

## Tested machines

So far, the package has been built and tested on the following platforms:

- Tomo3D 2.0 (January 2015) - x86\_64 (64 bits) and i686 (32 bits) under Linux
- Tomo3D 1.3.4 (April 2012) - Intel-based Macs under OSX 10.6.8
- Tomo3D 1.3.4 (April 2012) - 64-bit/32-bit machines under Windows (XP and onwards)

In case of problems, or needs for availability for other platforms, contact us and we'll try our best to sort them out.

---

<sup>1</sup>See Section devoted to Tomo3D under Windows for detailed information.

## 2 Description

Current computers usually ship with one or more multicore processor chips, every one comprising several powerful cores (e.g. dual-2, quad-4, hexa-6). Moreover, the individual cores contain vector instructions, typically known as SIMD (Single Instruction, Multiple Data), which perform multiple computations with different data elements of the same kind simultaneously. In particular, the SSE instructions (Streaming SIMD Extensions), present in all Intel and AMD processors, can perform four floating-point operations of the same type (addition, multiplication, etc) with their corresponding four pairs of operands. Since 2011, Intel and AMD provide the Advanced Vector eXtensions, which are vector instructions that double the number of simultaneous operations compared to SSE (i.e. eight simultaneous floating-point operations). All these features turn current computers into extremely powerful platforms with two levels of parallelism available for the programmer: threads and SSE/AVX instructions to exploit cores and vector units within the cores, respectively.

The 3D reconstruction problem in ET can be decomposed into a set of independent 2D reconstruction sub-problems corresponding to the slices perpendicular to the tilt axis. This has traditionally been the data decomposition used in ET for high performance computing. Each of the 2D slices of the volume can then be computed with any of the reconstruction methods, either WBP or SIRT but now working in 2D, from the corresponding set of 1D projections. The reconstruction of the independent slices can be carried out in parallel by using the different parallelism levels within the multicore computers.

Tomo3D uses a multithreaded vectorized approach to tomographic reconstruction (Figure 1). It uses the Single Program, Multiple Data (SPMD) paradigm whereby the different threads essentially execute the same task over different data. The program creates a number of threads, typically as many as cores in the computer, that run in parallel. The program keeps a pool with the slices to reconstruct, clustered in slabs of 4 or 8 slices (if SSE or AVX, respectively, are to be used). The slabs are asynchronously dispatched to the threads as soon as these are idle. When a slab is allotted to a thread, its four/eight slices are reconstructed simultaneously thanks to the SSE/AVX instructions, respectively. Once a thread finishes the reconstruction of the slab, it requests another one. The fact that slabs are dispatched on demand provides an inherent load-balancing mechanism and thus makes Tomo3D flexible to adapt to situations where the computer is also used by other users or programs.

Tomo3D has been implemented in C using Pthreads and SSE/AVX intrinsics. Single-core code optimization techniques have also been used to better exploit the memory hierarchy and the internal processing units within the cores. Memory consumption has been reduced to a minimum, only keeping a small subset of slabs and precomputed tables (e.g. cosines, etc.) used throughout the reconstruction process. I/O operations have been optimized to reduce disk latencies. Moreover, they are overlapped with computation, which succeeds in further reducing the effective program time. On the other hand, Tomo3D allows the use of two different hard disks so that the reading of the input tilt-series and the writing of the volume can run in parallel (see examples in Section 5). This will avoid interferences between readings and writings, which translates into a faster and smoother program execution.

Tomo3D is equipped with the two most common reconstruction methods in the field: WBP and SIRT. For the iterative method SIRT, a number of options are available: the positivity constraint that ensures that all the points in the reconstruction be positive, a weighting to com-

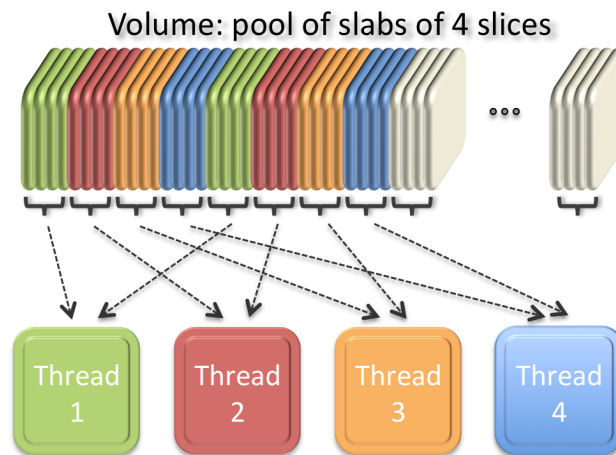


Figure 1: Multithreaded vectorized tomographic reconstruction in Tomo3D. A pool of slabs of 4 or 8 slices (if SSE or AVX, respectively, are to be used) is maintained. As many threads as cores run concurrently in the system. The slabs are asynchronously dispatched to the threads as soon as they are idle. The four/eight slices in a slab are reconstructed simultaneously thanks to SSE/AVX vector instructions, respectively. In the figure, allocation of slabs to threads are colour-coded, and the slabs are of 4 slices.

penstate the "long object effect" (Xu et al. *J.Struct.Biol.* 171:142–153, 2010) and the possibility of resuming a reconstruction with more iterations. Tomo3D reads tilt-series and writes the output tomograms in MRC format and is fully compatible with standard ET packages (e.g. IMOD). Tomo3D is provided as a stand-alone, download-and-go, program that does not require any special hardware or software library in the system, which makes it especially attractive for structural biology laboratories.

Detailed description of the procedures implemented in the program as well as performance results have been published in several articles. Please visit the References Section at the end of this document for a list of these articles and links to them.

If you use Tomo3D in your works, please cite any of the following articles:

[Tomo3D 2.0 - Exploitation of Advanced Vector eXtensions \(AVX\) for 3D reconstruction.](#)

J.I. Agulleiro, J.J. Fernandez

*J. Struct. Biol.*, 189:147–152, 2015.

[Fast tomographic reconstruction on multicore computers.](#)

J.I. Agulleiro, J.J. Fernandez

*Bioinformatics* 27:582–583, 2011.

### 3 Usage

The program works with a command-line user interface that adheres to typical Unix-style practices. The format of the input tilt-series is MRC, with several modes accepted (0 - unsigned char; 1 - short; 2 - float; 6 - unsigned short), as typically used with standard packages like IMOD. The file containing the tilt angles is a text file in IMOD style (i.e. a number per line, where the numbers represent the tilt angle). The format of the output tomogram is MRC, mode 2 (i.e. single precision floating point numbers).

To get the list of options, just type the program name followed by ‘-h’: `tomo3d -h`  
It will show the following information on console:

```
General usage: tomo3d -a tiltangs_file -i tiltseries
```

Optional parameters:

```
-A offset      Add offset (a real number) to each tilt angle (default: 0)
-b n,m        Size of I/O buffers (n=No. of sinograms, m=No. of slices)
              (default: auto)
-C n          Process slices and sinograms using a cache block of n KB
              (default: auto)
-c            Constrain output volume
-d            Display system information and exit
-e avx/sse/none Select vector extensions (default: auto)
-f            Silently overwrite output volume
-H            Use Hyper-Threading if available
-h            Show this help and exit
-I on/off     Turn on/off interpolation (default: on)
-L            Reconstruct from the base 10 logarithm of input tiltseries
-l iter       Iterations for the Iterative Reconstruction Methods
              (default: 30)
-M memlimit   Impose a limit on how much RAM is used (default: 0.75 =
              75%)
-m freq       Frequency ([0,0.5]) from which Hamming filter is applied in
              WBP (default: 0)
-n            Change the handedness of the tomogram (default: IMOD style)
-O densoffset Offset each sinogram pixel using density 'densoffset'
              (default: 0)
-o tomogram   Name output reconstructed volume
-P n          Sinograms will be processed in sets of n projections
              (default: auto)
-R n          Slices will be processed in sets of n rows (default: auto)
-r tomogram   Resume a previous SIRT reconstruction
-S            Select SIRT (default: WBP)
-s splitf     Factor to split slice rows (default: auto)
-t threads    Use threading (default number of threads: auto)
-V            Display version information and exit
-v level      Set verbosity level: 0 (no verbose), 1 (low, default
              level), 2 (high)
-w on/off     Turn on/off weighting (default: on)
-x width      Focus reconstruction instead of using default width
-y y1,y2      Focus reconstruction instead of using default depth
-z height     Focus reconstruction instead of using default height
```

Note that the options are case-sensitive. The options ‘-a’ and ‘-i’ are mandatory. They indicate the name of the files containing the tilt angles and the tilt series, respectively.

## 4 Options and Parameters

### Mandatory parameters

**-i <tiltseries>**

Name of the file containing an aligned tilt series. The format of the tilt series has to be MRC. Any mode (0 - unsigned char; 1 - short; 2 - float; 6 - unsigned short) is accepted, as typically used with standard packages like IMOD. The tilt axis is assumed to be along the Y axis.

**-a <tiltangs\_file>**

Name of the ASCII file containing the tilt angles of the images in the tilt series. There must be one tilt angle per line. The order of the tilt angles in this file must correspond to the order of the images in the input tilt series. The format of this file is the same as that used by IMOD.

### Optional parameters for the general user

**-A <offset>**

This option allows the user to apply an offset to the tilt angles. The result is a rotation of the tomogram around the tilt axis. The user might want to use this option to make the slab reconstruction look horizontal. In combination with the options -x, -y, -z, it helps to focus the reconstruction to an area of interest and further reduce the processing time.

This option should be used with caution if subtomogram averaging is to be carried out because the missing wedge also undergoes a rotation around the tilt axis by this offset. So, the user should bear this in mind when the missing wedge information is input to subtomogram averaging packages.

**-d**

*– Only available in Tomo3D 2.0 –*

Flag to make the program show information about the features of the processors detected in the computer (model, N. of cores, if Hyper-Threading is supported, size of L1 and LLC cache memories, vector instructions that are available, etc.). The processor denoted by “-1” is a virtual one and summarizes the whole computational power available in the computer. Tomo3D uses this information for autoconfiguration. See Section 5.2 (Examples of the output report) for a sample output.

**-c**

Flag to activate the constraining of the output volume. By default, it is off. This option switches on the positivity constraint, which may be especially useful for SIRT. At each iteration, it ensures that the densities of the voxels in the output volume be positive, setting those negative densities to zero.

**-e <avx/sse/none>**

*– Available in Tomo3D 2.0 –*

**-e <on/off>**

*– Available in Tomo3D 1.3.4 –*

This option allows the activation/deactivation of vector instructions, and has different syntax in Tomo3D 2.0 and 1.3.4.

In Tomo3D 2.0, this option allows the activation (avx/sse) or deactivation (none) of the use of vector instructions (AVX/SSE). By default, Tomo3D automatically detects their presence. If the processor supports AVX instructions, they will be selected. If not, Tomo3D will employ SSE instructions, if available. With this option of Tomo3D, the user may force the use of SSE even though AVX are supported. Deactivation of vector instructions (none) would make the program work much slower.

In Tomo3D 1.3.4, the user only has the option of activating (on) or deactivating (off) the SSE instructions. By default, they are used if the processor supports them.

In general, it is not recommended to disable vector instructions because, at least, SSE are present in the vast majority of Intel and AMD processors, and Tomo3D takes advantage of them.

**-f**

Flag to silently overwrite the output volume if it exists. The program checks the existence of a file with the name indicated by the user (see option -o). If it exists, by default the program shows a comment on console and asks the user whether it must be overwritten or not. This option allows the program to skip this control.

**-H**

The program automatically detects whether the cores have hyperthreading technology (e.g. Intel Nehalem architecture: Core i7, i5, i3, etc.). However, it does not exploit this feature by default. 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. Nevertheless, bear in mind that in an application like Tomo3D 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 reached by two real cores working in parallel. According to our tests, mixing AVX and hyperthreading is discouraged. When using SSE, a speedup between 10% and 30% can be obtained in SIRT thanks to hyperthreading, but we have not detected a real gain in WBP, though. Anyway, it is recommended to experiment with this option and see if it is good for your system. This flag is ignored if the option -t is used.

**-h**

The program shows the help information on console and exits.

**-I <on/off>**

This option allows activation/deactivation of linear interpolation in the projection/backprojection processes involved in WBP and SIRT. By default, linear interpolation is used (on), as it provides better results, with apparent visual appeal. If it is off (i.e. nearest neighbour, instead), the program runs faster, but at the expense of poorer results.

- L** *– Only available in Tomo3D 2.0 –*  
 Flag to compute the tomogram from the base 10 logarithm of the input tiltseries. By default, the logarithm is not applied, assuming that the pixel values in the input tiltseries are proportional to the projected mass density. The program automatically adds a density offset before applying the logarithm to force positive pixel values in the output tiltseries. The option -O could also be used to manually input this offset.
- l <iterations>**  
 It sets the number of iterations for SIRT. By default, a number of 30 iterations is used as long as the SIRT method is selected (see option -S). If WBP is chosen, this option does not have any effect.
- M <memlimit>** *– Only available in Tomo3D 2.0 –*  
 This option allows the user to limit the amount of RAM memory used by Tomo3D. By default, the program will never require more than 75% of the RAM memory available in the computer. This limit might be exceeded only in the unlikely event that the tomogram is exceptionally huge for the machine. The argument is entered as a fraction (i.e. input of 0.75 is understood as 75% of the memory). In combination with -t, this option may be useful to keep Tomo3D as a light process in the machine, without excessive resource consumption. The use of this option may affect the size of the I/O buffers because they might have to be modified to fulfill the input memory limit.
- m <freq>**  
 By default, in WBP the program applies a Ramp filter together with a Hamming filter. The latter allows attenuation of the high frequency components, which are typically very noisy. This parameter allows setting up the starting frequency from which the Hamming filter is to be applied. The values must range in [0,0.5], where 0.5 represents the Nyquist frequency. By default, the program uses a starting frequency of 0, i.e. a Hamming is applied over the whole frequency range. A value of 0.5 would turn off the Hamming filter. Values in-between would preserve low-frequency components and modulate the contribution of the other frequency components (see Section on Advices and comments for more details).
- n**  
 Flag to change the handedness of the tomogram. By default, the program writes the output tomogram as IMOD. This option makes the program write it upside down.
- O <densoffset>** *– Only available in Tomo3D 2.0 –*  
 This option allows the user to specify a density value that will be added to all pixels in the tiltseries before reconstruction. This might be used to force pixels of the input tiltseries to positive values so that the positivity constraint make sense.
- o <tomogram>**  
 This parameter is the name of the output file where the reconstructed tomogram is to be saved. The format is MRC, mode 2 (floating-point). The tomogram has the output slices perpendicular to the plane of the specimen, as this is the way in which the slices are computed (i.e. similarly to the way IMOD works by default). If this parameter is not entered, the program does save the reconstructed tomogram on a file whose name is built by adding the prefix “recon\_” to the name of the input tilt series.



**-r <tomogram>**

This option allows the user to resume a SIRT reconstruction with more iterations. The parameter is the name of the file containing a tomogram previously computed with Tomo3D using SIRT. This tomogram will be used as a starting point for the current execution of Tomo3D.

If the options -x, -y, -z are used in the computation of the input tomogram, the same values of width, slices and thickness (which are stored in the header of the input tomogram) will keep for the resumed reconstruction. Even so, the user may still use these options. In that case, the values input with -x and -z must match the input tomogram. Regarding the slices, the option -y might be used to further reduce the size of the tomogram.

**-S**

Flag to activate the tomographic reconstruction with SIRT. By default, Tomo3D uses WBP. This option allows the user to set SIRT as the reconstruction method.

**-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 reconstruction 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).

**-V**

Shows the version of the program and exits.

**-v <verbosity level>**

Sets the verbosity level: 0 (no verbose), 1 (low, default level), 2 (high). The verbosity level 2 only makes sense in SIRT, where the correlation between the experimental and calculated projections are shown on console slice-by-slice.

**-w <on/off>**

This option sets on/off the different weighting functions in the reconstruction methods. By default, the option is on, whatever method (WBP or SIRT) is selected.

For WBP, this option (on) refers to the use of the ramp filter. A Hamming filter is applied as well (see -m) so as to reduce the noisy contribution of high frequencies. If the weighting is off, the output tomogram is the result of the backprojection process, without weighting.

For SIRT, the weighting (on) refers to the “long object compensation” described by Xu et al., J.Struct.Biol. 171:142–153, 2010.

**-x <width>**

By default, the width of the reconstructed tomogram is equal to the X dimension of the images in the tilt-series. This option allows the user to set the width of the tomogram to a smaller value, which helps to better focus the reconstruction to an area of interest and further reduce the processing time. Regardless of the width value, the tomogram is always reconstructed around the tilt axis (i.e. the centre of the images in the tilt-series).

**-y <y1,y2>**

By default, the output tomogram has as many slices along the tilt axis as the Y dimension of the images in the tilt-series. This option allows the user to specify the set of slices to be reconstructed. Thus, this option helps to better focus the reconstruction to an area of interest and further reduce the processing time. The user has to specify two values y1,y2 numbered from 0 (i.e. in the range [0,Ny-1], with Ny being the Y dimension of the images), which represent the indices of the first and last slices to be included in the output tomogram.

**-z <height>**

By default, the height (i.e. thickness) of the reconstructed tomogram is equal to the X dimension of the images in the tilt-series. This option allows the user to properly set the height of the tomogram to a value that better fits the thickness of the specimen under study. This also allows reduction of the processing time. The thickness value has to be an even number. This is necessary to exploit projection symmetry (see Agulleiro et al. J.Struct.Biol. 170:570–575, 2010).

## Optional parameters for the expert user

### **-b <n,m>**

This parameter sets the size of the I/O buffers used to read/write the tilt-series/tomogram from/to disk.  $n$  denotes the size (number of sinograms) of the input buffer.  $m$  is the size (number of slices) of the output buffer. These parameters may have an important influence on the memory consumption and the performance of the I/O of the program. Higher values involve higher memory consumption but, in general, the performance improves, especially for WBP.  $n$  and  $m$  must not be higher than the total number of sinograms and slices, respectively. If AVX or SSE instructions are used, the input values must be multiple of eight or four, respectively.

In Tomo3D 2.0, optimal values are automatically calculated so that large I/O buffers are used within reasonable memory consumption. In Tomo3D 2.0, use of this option overrides whatever input with the option -M.

In Tomo3D 1.3.4, the default values for  $n$  and  $m$  are fixed to 128,128.

### **-C <n>**

In order to better exploit the memory hierarchy, the program processes the sinograms in sets of  $p$  projections and the slices in sets of  $r$  rows. By default, the program automatically detects the cache memory available in the computer and optimizes the values of  $p$  and  $r$  accordingly. This option allows the user to specify a particular size of the cache block to store the  $p$  projections and the  $r$  rows during the processing. The value to input denotes a cache block of  $n$  KB and should not be larger than the LLC (Last Level Cache) cache memory available in the processor. This option is incompatible with the options -P and -R.

### **-P <p>**

The program processes the sinograms in sets of  $p$  projections in order to better exploit the memory hierarchy (see Agulleiro et al. J.Struct.Biol. 170:570–575, 2010). By default, the program fixes this parameter based on the cache memory available in the computer, which is determined automatically. This option allows the user to specify a particular value (see also -C). As a general rule, those which are small power-of-two are recommended. The input value must not be higher than the number of images in the tilt series. Setting this parameter is tricky and may have a striking (good or bad) effect on the reconstruction speed.

### **-R <r>**

The program reconstructs the slices in sets of  $r$  rows in order to better exploit the memory hierarchy (see Agulleiro et al. J.Struct.Biol. 170:570–575, 2010). By default, the program fixes this parameter based on the cache memory available in the computer, which is determined automatically. When SSE are enabled, this option allows the user to specify a particular value (see also -C). As a general rule, those which are small power-of-two are recommended. If AVX are to be used, -R cannot be modified and always takes 1 as an argument. The input value must not be higher than half the thickness of the tomogram. Setting this parameter is tricky and may have a striking (good or bad) effect on the reconstruction speed.

**-s <splitf>**

*– Only available in Tomo3D 2.0 –*

This option denotes the split factor for the rows of the slices. This is essential for processing with AVX instructions since it allows benefit of the L1 cache memory of the processor (Agulleiro and Fernandez, 2014). This parameter is disabled (it is set to 1) by default if SSE instructions are to be used. If AVX instructions are used instead, by default it is automatically calculated so that the pieces of the rows fit in less than half the L1 size. In particular, the split factor is fixed according to the formula  $L1\_size/(2*cores\_sharing\_L1)$ , where `cores_sharing_L1` denotes the number of real cores, or logical cores if Hyper-Threading is enabled. If this parameter is to be entered manually by the user, only values that are divisors of the size of the rows are acceptable.

## **Summary: The most important parameters**

The most important parameters are:

For reconstruction: the parameters to specify the input data, both tilt-series (-i) and angle list (-a), the output tomogram (-o) and the dimensions of the output tomogram (-x,-y,-z):

**-a, -i, -o, -x, -y, -z**

For WBP: the parameters to define the frequency range where the Hamming filter is to be applied (-m).

**-m**

For SIRT: the parameters to set SIRT as the reconstruction method (-S), indicate the number of iterations (-l) and, possibly, resume a previous SIRT reconstruction with more iterations (-r), or apply some constraints (-c).

**-S, -l, -r, -c**

## 5 Examples

### 5.1 Examples of commands

1. Reconstruction of a tomogram of thickness 256 voxels with WBP with the Hamming filter applied over the whole frequency range. The program works at full speed, i.e. internally it creates as many threads as cores available in the computer, which work in parallel collaborating in the reconstruction. Furthermore, all the threads take advantage of SSE instructions and thus reconstruct groups of 4 slices simultaneously. This way, the program takes the most of the power within the computer. All examples hereinafter work at full speed, unless otherwise stated.

```
tomo3d -a ang.tlt -i tseries.mrc -o tomogram.mrc -z 256
```

2. As above, but the Hamming filter is applied in the range [0.25,0.5]. This way, all the frequencies in [0,0.25] are preserved whereas those in [0.25,0.5] are progressively attenuated following the Hamming filter.

```
tomo3d -a ang.tlt -i tseries.mrc -o tomogram.mrc -z 256 -m 0.25
```

3. Reconstruction of a tomogram of thickness 200 voxels with 30 iterations (i.e. the default number of iterations) of SIRT at full speed. The positivity constraint is imposed at each iteration.

```
tomo3d -a ang.tlt -i tseries.mrc -o tomogram.mrc -z 200 -S -c
```

4. Reconstruction of a tomogram of thickness 200 voxels and width of 1400 voxels with 50 iterations of SIRT at full speed. Only a subset of 1000 slices along the tilt axis are to be reconstructed (from 600 to 1599). An offset of 10 degrees is added to the tilt angles so that the reconstruction in this example looks horizontal.

```
tomo3d -a ang.tlt -i tseries.mrc -o tomogram.mrc -x 1400 -y 600,1599  
-z 200 -A 10 -S -l 50
```

5. The previous reconstruction is to be run on a computer based on a Intel Core i7, which is a quad-core processor with hyperthreading technology available. The previous command would 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.

```
tomo3d -a ang.tlt -i tseries.mrc -o tomogram.mrc -z 200 -S -l 50 -H
```

6. 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. Although tomo3d adapts well to this situation, 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, the SSE instructions are still used internally.

```
tomo3d -a ang.tlt -i tseries.mrc -o tomogram.mrc -z 200 -S -l 50 -t 3
```

7. Reconstruction of a tomogram of thickness 180 voxels with 40 iterations of SIRT at full speed. The program runs with maximum level of verbosity in order to supervise the correlation between experimental and calculated projections for all the slices of the volume.

```
tomo3d -a ang.tlt -i tseries.mrc -o rec3d.mrc -z 180 -S -l 40 -v 2
```

8. Reconstruction of a tomogram of thickness 150, width 1000 and a number of 1200 slices (from 512 to 1711) along the tilt axis. Thirty iterations of SIRT at full speed are used. Later, if the user feels that more iterations of SIRT (let's say 20) would have been better, the reconstruction can be resumed. The program reads from the header of the input tomogram all the data concerning thickness, width and slices required to resume the reconstruction:

```
8.1. tomo3d -a ang.tlt -i tseries.mrc -o tomogram30.mrc -z 150 -x 1000
      -y 512,1711 -S -l 30
      (... inspection by the user ...)
8.2. tomo3d -a ang.tlt -i tseries.mrc -o tomogram50.mrc -S -l 20
      -r tomogram30.mrc
```

The two previous commands would be equivalent to the following one:

```
tomo3d -a ang.tlt -i tseries.mrc -o tomogram50.mrc -z 150 -x 1000
      -y 512,1711 -S -l 50
```

9. Reconstruction of a tomogram of thickness 180 voxels with 40 iterations of SIRT without using the “long object compensation”.

```
tomo3d -a ang.tlt -i tseries.mrc -o rec3d.mrc -z 180 -S -l 40 -w off
```

10. Reconstruction of a tomogram of thickness 150 voxels with 50 iterations of SIRT using the nearest-neighbour algorithm instead of linear interpolation.

```
tomo3d -a ang.tlt -i tseries.mrc -o rec3d.mrc -z 150 -S -l 50 -I off
```

11. Reconstruction of a tomogram of thickness 512 voxels with WBP using two different hard disks, one for reading the input tilt-series and the other for writing the volume. This further reduces the processing time as reading and writing run in parallel without interference to access the disk. To do this, the input tilt-series needs to be stored in one of the disks and the path of the volume must belong to the other. First example refers to Linux and OSX, while the second one is for Windows.

```
11.1. tomo3d -a ang.tlt -i /hd1/tseries.mrc -o /hd2/rec3d.mrc -z 512
```

```
11.2. tomo3d -a ang.tlt -i c:\tseries.mrc -o d:\rec3d.mrc -z 512
```

## 5.2 Examples of the output report

1. On the execution of 30 iterations of SIRT on a quad-core computer based on Intel Core 2 Quad Q6600 at 2.4GHz with the following command using verbosity level 1 (the default):

```
tomo3d -a ang.tlt -i tseries.mrc -o rec3d.mrc -z 256 -S -l 30
```

the following output report is shown on console:

```
Input dataset:      tseries.mrc
Output recon file:  rec3d.mrc
Running SIRT (30 iterations).
I/O buffers set to 128/128.
SSE enabled.
4 thread(s) will be created.
Output volume will be 1024 x 256 x 1024 (1.0 GB).
100.0% completed (RAM usage: 183.1MB/7.5GB).
Reconstruction time = 198.8730 sec.
I/O time = 2.9914 sec.  {read = 0.2309, write = 2.7605, non-overlapped = 0.8690}
Program execution time = 200.1470 sec.
Reconstruction successful!
```

This report shows on console the information related to the reconstruction method, the size of the volume, the capabilities of the platform being exploited (e.g. SSE and/or threads), the memory consumption and the processing time. The line “I/O buffers set to 128/128” informs about the size of I/O buffers selected by the program (or the user) and is only shown in Tomo3D 2.0.

If the verbosity level 2 is used, the correlation between experimental and calculated projections for all the slices of the volume is shown.



2. If Tomo3D 2.0 is invoked with the option `-d` (`tomo3d -d`), information about the features of the processors detected in the computer are shown on console, as in the following example:

```
Processor           : 0
Model name          : Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
Vendor ID           : GenuineIntel
Cores in chip       : 4
Hardware threads/core : 2
Logical cores       : 8
Cache size          : 8192 KB
Cache shared by     : 4 cores, 8 logical
L1 cache size       : 32 KB
L1 shared by        : 1 core, 2 logical
Features            : sse avx ht
```

```
Processor           : -1
Model name          : Tomo3D Virtual CPU
Vendor ID           : TrueTomo3D
Cores in chip       : 4
Hardware threads/core : 2
Logical cores       : 8
Cache size          : 8192 KB
Cache shared by     : 4 cores, 8 logical
L1 cache size       : 32 KB
L1 shared by        : 1 core, 2 logical
Features            : sse avx ht
```

This system only comprises one processor chip (the one represented by *Processor: 0*). The processor in question is a Core i7-2600 from Intel (see fields *Model name* and *Vendor ID*) and has 4 real cores as indicated by *Cores in chip*. Since it supports Hyper-Threading (as denoted by 'ht' in the *Features* field), it can run two threads per core (field *Hardware threads/core*), totalling 8 logical cores (*Logical cores*). This processor is equipped with an LLC of 8192 KB (*Cache size*), which is shared by 4 real cores and 8 logical ones (*Cache shared by*). It also owns a first level cache whose size is 32 KB (*L1 cache size*). This cache is shared by 1 real core and 2 logical ones (*L1 shared by*). The processor supports SSE and AVX vector instructions, as denoted by 'sse' and 'avx', respectively, in the *Features* field.

The processor represented by -1 is a virtual one and summarizes all the computing power available in the system. Since there is only one processor chip installed, processor -1 is a copy of processor 0.

## 6 Advices and comments

- *Tuning the Hamming filter.*

In WBP Tomo3D applies the very well known Ramp filter and also a Hamming filter. The latter is intended to attenuate the noise typically contributed by the high frequency components. In Tomo3D we have included a parameter (`freq`, option `-m`) that allows the user to set up the starting frequency from which the Hamming filter is applied. All the frequencies lower than `freq` are subjected to the Ramp filter only. For frequencies higher than `freq`, the attenuation of the components takes place through the combination of the Ramp and Hamming filters. Figure 2 shows the resulting combination Ramp+Hamming filter for different values for this starting frequency `freq`.

⇒ If subtomogram averaging is planned, please use values of `freq` in the range  $[0.35, 0.5]$ . These values ensure good preservation of the signal for the alignment process.

⇒ In case of electron tomography studies where subtomogram averaging is not planned, values of `freq` in the ranges  $[0, 0.15]$  or  $[0, 0.25]$  are helpful to reduce the noise and improve the contrast, thereby facilitating the subsequent interpretation stages (e.g. denoising with more sophisticated methods; segmentation; etc.).

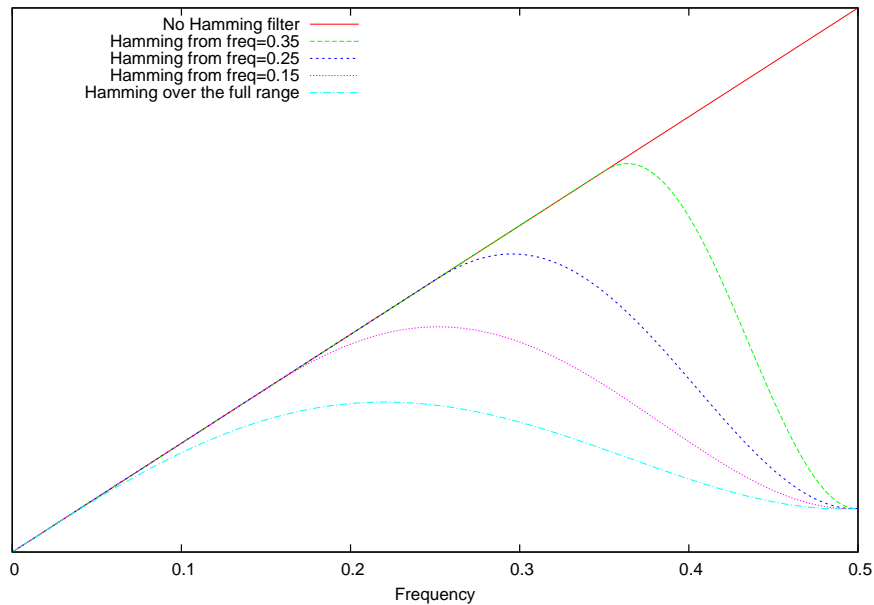


Figure 2: The actual filter applied in WBP by Tomo3D for different values of the starting frequency for the Hamming filter. With `freq=0.5`, the classical well known Ramp filter is applied over the whole frequency range (red), and the Hamming filter is not applied at all. With `freq=0`, the Hamming filter is applied in addition to the Ramp filter at all frequencies (cyan). At different values of `freq` (0.15, 0.25, 0.35 in the plot), the standar Ramp filter is applied up to the precise value of `freq`. From that value on, the Hamming filter is applied as well. The higher the value of `freq` is, the attenuation starts at higher frequencies.

## 7 Tomo3D under Windows

Tomo3D is a program that runs on command line terminals. Under Windows, Tomo3D runs on Command Prompt, which is accessible through Start+Accessories+Command Prompt. At the moment, only Tomo3D 1.3.4 is available for Windows.

The reader can find more information about Windows Command Prompt in the following links:

[http://en.wikipedia.org/wiki/Command\\_Prompt](http://en.wikipedia.org/wiki/Command_Prompt)

[http://en.wikipedia.org/wiki/List\\_of\\_MS-DOS\\_commands](http://en.wikipedia.org/wiki/List_of_MS-DOS_commands)

<http://www.bleepingcomputer.com/tutorials/windows-command-prompt-introduction/>

The user should set up the PATH so that the program Tomo3D is available automatically on the terminal. Alternatively, just copy the executable program Tomo3D to the location where the datasets are. An explanation about how to set the PATH under Windows is out of the scope of this documentation. The following links may be of some help:

<http://vlaurie.com/computers2/Articles/environment.htm>

<http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/path.msp?mfr=true>

On Command Prompt, the user should go to the directory where the datasets are (with the command 'cd'). Then, the tomograms can be computed with Tomo3D as described in the other sections of this documentation. However, the user should bear in mind the following points:

- For Windows XP (Service Pack 3) and beyond (Windows Vista, Windows 7, ...), the full functionality of Tomo3D is available, and it will work exactly as described here.
- For other Windows XP (original, Service Pack 1 and 2), the automatic detection of the number of cores and cache size does not work. Tomo3D warns the user about that. He could then use the options '-t' and '-C' to manually tune these parameters. Therefore, under Windows XP, the full functionality of Tomo3D is available as long as the Service Pack 3 is installed. It can be downloaded from the following link:

<http://www.microsoft.com/download/en/details.aspx?id=24>.

## 8 History

### V. September 2010

- Original package.

### V. April 2011

- The program automatically detects the number of cores and whether hyperthreading is available.
- By default, the program fully exploits the computing power in the computer by using SSE instructions and creating as many threads as cores.
- The program automatically detects the size of the L2 cache memory and accordingly tunes the parameters R and P in an optimal way.
- I/O operations are overlapped with computation.
- The Hamming filter is available in WBP.
- A parameter to change the handedness of the tomogram is now available.
- More user-friendly parameters for experts to tune the performance of the program.
- Additional minor optimizations have also been applied.

### V. April 2012 – Tomo3D 1.3.4

- Major code re-organization in order to make the same version of the program run under Linux, OSX and Windows.
- The new options -A, -x, -y are available. These, in combination to -z, help to focus the reconstruction to an area of interest, reduce the size of the output tomogram and reduce the processing time.
- The use of the option '-r' does not require to specify the thickness of the input tomogram any more. In this version, all the data concerning thickness (-z), width (-x) and slices (-y) of the input tomogram are stored in its header.
- The information on the pixel size and sampling is read from the input tilt-series and passed on to the output tomogram.
- Tomo3D can run on old processors not supporting SSE instructions (e.g. a Pentium).

### V. January 2015 – Tomo3D 2.0

- Major code re-organization to exploit AVX vector instructions.
- Tomo3D now can deal properly with FEI stacks and accept stacks with MODE 6 of MRC format.
- The I/O buffers are optimally calculated to reduce turn-around time and to make an efficient use of the computer memory.
- The new options -d, -L, -M, -O, -s are available.
- Tomo3D has been updated to work with MRC labels.
- Fixed a bug when printing correlations if SSE were enabled: if the number of sinograms to process was not a multiple of 4, some garbage correlations were printed.
- Functions to pack/unpack slabs have been optimized. As a result, WBP using SSE is now slightly faster than the previous version implemented in Tomo3D 1.3.4.

## **Acknowledgements**

Dr. S Li, Dr. M Beeby, Dr. KH Bui and Dr. FJ Chichon provided useful comments from the user's point of view. Help and support from Dr. EM Garzon and Dr. I Garcia are gratefully acknowledged. This work has been possible thanks to grants from the Spanish MEC and MCI, Junta de Andalucia and CSIC.

## References

### References describing the implementation in Tomo3D

[Tomo3D 2.0 - Exploitation of Advanced Vector eXtensions \(AVX\) for 3D reconstruction.](#)

JI Agulleiro, JJ Fernandez.

*J. Struct. Biol.*, 189:147–152, 2015.

[Tuning the cache memory usage in tomographic reconstruction on standard computers with advanced vector eXtensions \(AVX\).](#)

JI Agulleiro, JJ Fernandez.

*Data in Brief*, (in press), 2015. doi: 10.1016/j.dib.2014.12.010

[Evaluation of a multicore-optimized implementation for tomographic reconstruction.](#)

JI Agulleiro, JJ Fernandez.

*PLoS ONE* 7(11):e48261, 2012.

[Fast tomographic reconstruction on multicore computers.](#)

JI Agulleiro, JJ Fernandez.

*Bioinformatics* 27:582–583, 2011.

[Vectorization with SIMD extensions speeds up reconstruction in electron tomography.](#)

JI Agulleiro, EM Garzon, I Garcia, JJ Fernandez.

*J. Struct. Biol.* 170:570-575, 2010.

### References on the tomographic reconstruction methods implemented in Tomo3D

P Bruyant. [Analytic and Iterative Reconstruction Algorithms in SPECT.](#) *J. Nucl. Med.* 43:1343-1358, 2002.

P Gilbert. [Iterative methods for the 3D reconstruction of an object from projections.](#) *J. Theor. Biol.* 76:105-117, 1972.

GT Herman. [Fundamentals of Computerized Tomography.](#) Springer, 2010.

AC Kak, M Slaney, [Principles of Computerized Tomographic Imaging](#), Society of Industrial and Applied Mathematics, 2001.

M Radermacher. [Weighted Back-projection methods.](#) In *Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell*, (J Frank, ed.), 2006.

### References related to fast 3D reconstruction in electron tomography

JI Agulleiro, F Vazquez, EM Garzon, JJ Fernandez. [Hybrid computing: CPU+GPU co-processing and its application to tomographic reconstruction.](#) *Ultramicroscopy* 115:109–114, 2012.

D Castano-Diez, H Mueller, AS Frangakis. [Implementation and performance evaluation of reconstruction algorithms on graphics processors.](#) *J. Struct. Biol.* 157:288–295, 2007.

- D Castano-Diez, D Moser, A Schoenegger, Pruggnaller, AS Frangakis. [Performance evaluation of image processing algorithms on the GPU](#). *J.Struct.Biol.* 164:153–160, 2008.
- JJ Fernandez. [High performance computing in structural determination by electron cryomicroscopy](#). *J. Struct. Biol.* 164:1–6, 2008.
- JJ Fernandez, A Lawrence, J Roca, I Garcia, M Ellisman, JM Carazo. [High performance electron tomography of complex biological specimens](#). *J.Struct.Biol.* 138:6–20, 2002.
- JJ Fernandez, JM Carazo, I Garcia. [Three-dimensional reconstruction of cellular structures by electron microscope tomography and parallel computing](#). *J. Parallel Distrib. Comput.* 64:285–300, 2004.
- JJ Fernandez, D Gordon, R Gordon. [Efficient parallel implementation of iterative reconstruction algorithms for electron tomography](#). *J.Paral.Distrib.Comput.* 68:626-640,2008.
- WJ Palenstijn, KJ Batenburg, J Sijbers. [Performance improvements for iterative electron tomography reconstruction using graphics processing units \(GPUs\)](#). *J. Struct. Biol.* 176:250–253, 2011.
- F Vazquez, EM Garzon, JJ Fernandez. [A matrix approach to tomographic reconstruction and its implementation on GPUs](#). *J. Struct. Biol.* 170:146–151, 2010.
- F Vazquez, EM Garzon, JJ Fernandez. [Matrix implementation of simultaneous iterative reconstruction technique \(SIRT\) on GPUs](#). *The Computer Journal* 54:1861–1868, 2011.
- W Xu, F Xu, M Jones, B Keszthelyi, J Sedat, D Agard, K Mueller. [High-performance iterative electron tomography reconstruction with long-object compensation using graphics processing units \(GPUs\)](#). *J. Struct. Biol.* 171:142–153, 2010.
- SQ Zheng, B Keszthelyi, E Branlund, J Lyle, MB Braunfeld, JW Sedat, DA Agard. [UCSF tomography: An integrated software suite for real-time electron microscopic tomographic data collection, alignment, and reconstruction](#). *J.Struct.Biol.* 157:138–147, 2007
- SQ Zheng, E Branlund, B Keszthelyi, MB Braunfeld, Y Cheng, JW Sedat, DA Agard. [A distributed multi-gpu system for high speed electron microscopic tomographic reconstruction](#). *Ultramicroscopy* 111:1137–1143, 2011.

#### References on high performance computing and computer architecture related to Tomo3D

- N Fiasta, M Buxton, P Jinbo, K Nasri, S Kuo. [Intel AVX: New frontiers in performance improvements and energy efficiency](#). *Intel Software Solutions Group*, 2008.
- M Hassaballah, S Omran, YB Mahdy. [A review of SIMD multimedia extensions and their usage in scientific and engineering applications](#). *The Computer Journal*, 51:630–649, 2008.
- JL Hennessy and DA Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann. 2007.
- VW Lee *et al.*. [Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU](#). *SIGARCH Comput. Archit. News* 38(3):451–460, 2010.