ICON-GPU 1.2.5 User Guide

1. Installation

Before installation of ICON-GPU, please make sure that you have installed NIVIDA CUDA package (version 6.5 or higher) and configured the environment properly.

Go into the installation directory and run **./install** to install ICON-GPU.

The script **install** firstly extracts the fftw-3.3.4.tar.gz in the directory of **supportLib.** And then, it generates the static link library libfftw3f.a, and copy it into the directory of **lib**. And then, a configure file **logLocation.conf** will be created in the directory of **config**, **logLocation.conf** describes the location (default as the installation path) of the log file **ICONlog.txt**, user can change the location by modifying **logLaction.conf** and the location should be an absolute path**.** Finally, it generates 4 executable files in the directory of **bin**, including **ICONPreProcess**, **ICON-GPU, ICONMask1** and **ICONMask2.**

2. Demand of Device Memory

The CUDA program **ICON-GPU** needs 2.5 GB, 1 GB and 0.5 GB memory on GPU devices to reconstruct a slice with size of 4k*4k, 2k*2k and 1k*1k, respectively.

3. Usage of ICON-GPU

Notice: ICON-GPU only performs a full ICON reconstruction and a cross validation process, users should use **ICONPreprocess** and **ICONMask1/ICONMask2** (same with ICON) to preprocess the tilt series and combine the reconstructed slices.

✓ Tilt series preprocess using ICONPreProcess

This program preprocesses the projection file by two steps. Firstly, subtracting the mode value of each projection image. Secondly (optional), normalizing the variance of each tilt image to be 0.33*thickness/cos(tilt-angle).

The parameters are described as follows.

-input (-i) : the tilt series.

-tiltfile (-t) : the file containing aligned tilt angle of each projection image. If this option is not used, then only subtract the mode value of projection images.

-thickness (-th) : the thickness of specimen in pixel. If this option is not used, then only subtract the mode value of projection images.

-output (-o) : the preProcessed projection file.

-help (-h) : for help.

For example:

./ICONPreProcess **-input** test.ali **-output** preprocessed_test.ali

or

./ICONPreProcess **-input** test.ali **-tiltfile** test.tlt **-thickness** 100 **-output** preprocessed_test.ali

Attention: (1) It is recommended to run this step against the original tilt series before alignment but using the aligned tilt file. After preprocessing, you can run newstack in IMOD to generate a preprocessed and aligned tilt series. (2) The second step (normalization) is optional and will be executed only when '-thickness' and '-tiltfile' are not empty.

✓ 3D reconstruction using ICON-GPU

This program is compiled with CUDA and performs a full ICON reconstruction and a cross validation process at the same time using Graphics Processing Unit (GPU). Two folders named **crossValidation** and **reconstruction** will be created in the "**-outputPath**" (a parameter defined by user, see parameters description).

In the folder **crossValidation**, five files will be created including:

a.GroundTruth.mrc, the omitted projection image at the minimum tilt angle (the smallest abs value);

b.crossV_reProjection.mrc, the re-projection image of the reconstruction generated by cross validation process;

c. fullRec_reProjection.mrc, the re-projection image of the reconstruction generated by full ICON reconstruction;

d. crossV.frc, the FRC calculated between GroundTruth.mrc and crossV_reProjection.mrc;

e. fullrec.frc, the FRC calculated between GroundTruth.mrc and fullRec_reProjection.mrc**.**

Attention: (1) crossV.frc and fullrec.frc will be used in **ICONMask1** or **ICONMask2**. (2) For ICON-GPU 1.2.5, ICON-GPU can only reconstruct square tilt series with nx = ny, please clip the tilt series into square first.

In the folder **reconstruction,** a series of 2D full reconstruction slices (without mask) named

minxxxxx.mrc will be generated. Such MRC files will be combined and masked (in Fourier domain) to generate the final 3D reconstruction by **ICONMask1** or **ICONMask2** in the next step.

The parameters of **ICON-GPU** are described as below:

-input (-i) : the aligned tilt series.

-tiltfile (-t) : the aligned tilt file.

-outputPath (-o) : the path of a folder saving the result, two folder named “crossValidation” and “reconstruction” will be created inside.

-slice (-s) : the slices of reconstruction that include 2 parts split by ',' . For example, 0,511 means that reconstruct 512 slices ranging from slice 0 to slice 511.

-ICONIteration (-iter) : the iteration number including 3 parts split by ',' . For example, 5,50,10 means that, firstly, reconstruct using INFR for 5 iterations to generate a stable initial value, and then reconstruct using ICON for 50 iterations, and finally reconstruct using INFR for 10 iterations for fidelity.

-dataType (-d) : the type of dataset. There are two options: 1 for cryoET or plastic embedded ET (signal in black and background in white); 2 for negatively stained ET (signal in white and background in black); default as 1.

-threshold (-thr) : the threshold used in ICON, default as 0.03

-gpu (-g) : the gpu list used for calculation. For example, 0,2,4,6 means using four gpus: gpu 0, gpu 2, gpu 4 and gpu 6 for calculation. Default as -1, meaning automatically detecting the number of gpus and using all gpus in the system for calculation.

-help (-h) : for help

One example of running ICON-GPU using all gpus:

./ICON-GPU **–input** preprocessed_test.ali **–tiltfile** test.tlt **-outputPath** testFolder **-slice** 0,511 **-ICONIteration** 10,50,10 **-dataType** 1 **-threshold** 0 **–gpu** -1

Or

One example of running ICON-GPU using gpu 0 and gpu 1:

./ICON-GPU **–input** preprocessed_test.ali **–tiltfile** test.tlt **-outputPath** testFolder **-slice** 0,511 **-ICONIteration** 10,50,10 **-dataType** 1 **-threshold** 0 **–gpu** 0,1

✓ Verification filtering based on cross validation FRC (ICONMask1 or ICONMask2)

Two programs, **ICONMask1** and **ICONMask2**, can be chosen to combine all the 2D reconstruction slices from **ICON-GPU** and generate a final verification filtered tomogram by masking out the unfaithful restored information in Fourier domain. The radius of mask is calculated according to the files of crossV.frc and fullRec.frc, which are generated by **ICON-GPU**. Different filtering strategies are used in **ICONMask1** and **ICONMask2**. For **ICONMask1**, the filtering is operation in a large 3D volume with the same size of the final tomogram. For **ICONMask2**, the filtering is operated ion a series of sub-volumes and then all these sub-volumes will be combined into the final tomogram. **ICONMask2** is more robust and memory efficient and it is always recommended, especially for a large tomogram.

Notice: ICONMask1 in ICON-GPU 1.2.5 can only deal with a reconstruction of the same X Y Z, which means the number of slices should be the same as the X/Y size of a midxxxxx.mrc.

ICONMask1

The parameters of **ICONMask1** are described as followed:

-inputPath (-i): the folder that contains all 2D reconstructed slices (named midxxxxx.mrc), normally corresponding to the **reconstruction** folder generated by ICON-GPU.

-tiltfile (-t) :the aligned tilt file.

-output (-o) : the masked 3D reconstruction.

-slice (-s) : the reconstructed slices for combination including 2 parts split by ','. For example, 0,511 means that combining 512 slices ranging from slice 0 (mid00000.mrc) to slice 511 (mid00511.mrc).

-thickness (-th) : the thickness of the final masked 3D reconstruction in pixel.

-radius (-r) : the mask radius (in pixel) used in the Fourier domain of the combined 3D reconstruction. If this option is used, 'crossVfrc' and 'fullRecfrc' are not used.

-crossVfrc (-cf) : the FRC curve from the cross validation process. If 'radius' is used, this option is not used.

-fullRecfrc (-ff) : the FRC file from the full reconstruction process. If 'radius' is used, this option is not used.

-help (-h) : for help

For example:

./ICONMask1 **-inputPath** testFolder/reconstruction **-tiltfile** test.tlt **-output** masked_ICONreconstruction.mrc **-slice** 0,511 **-thickness** 512 **-crossVfrc** testFolder/crossValidation/crossV.frc **-fullRecfrc** testFolder/crossValidation/fullRec.frc

ICONMask2

The parameters of **ICONMask2** are described as followed:

-inputPath (-i) : the folder that contains all 2D reconstructed slices (named midxxxxx.mrc), normally corresponding to the **reconstruction** folder generated by ICON-GPU.

-tiltfile (-t) :the aligned tilt file.

-output (-o) : the masked 3D reconstruction.

-slice (-s) : the reconstructed slices for combination including 2 parts split by ','. For example, 0,511 means that combining 512 slices ranging from slice 0 (mid00000.mrc) to slice 511 (mid00511.mrc).

-thickness (-th) : the thickness of the final masked 3D reconstruction in pixel.

-radius (-r) : the mask radius (in pixel) used in the Fourier domain of the combined 3D reconstruction. If this option is used, 'crossVfrc' and 'fullRecfrc' are not used.

-gaussWidth (-gw) : the width of gaussian edge of the soft mask (in pixel). If '-crossVfrc' & '-fullRecVfrc' are used, the default value is calculated accroding to FRC0.3 - FRC0.5; if '-radius' is used, the default value is 10.

-crossVfrc (-cf) : the FRC curve from the cross validation process. If 'radius' is used, this option is not used.

-fullRecfrc (-ff) : the FRC file from the full reconstruction process. If 'radius' is used, this option is not used.

-zshift (-z) : the shift (in pixel) of sample in Z axis, default as 0.

-nomask (-nm) : if this value is set to 0 then a validation filtering mask will be executed; otherwise, no validation filtering mask will be executed, default as 0.

-blockSize (-bs) : the size of sub-volume (a cube mask), default as 150.

-help (-h) : for help

For example:

```
./ICONMask2 -inputPath testFolder/reconstruction -tiltfile test.tlt -output masked_ICONreconstruction.mrc -slice 0,511 -thickness 200 -crossVfrc testFolder/crossValidation/crossV.frc -fullRecfrc testFolder/crossValidation/fullRec.frc
```

4. Citation of ICON-GPU

1. Chen Y., Wang Z., Zhang J., Li L., Wan X., Sun F.* and Zhang F.* (2017), Accelerating electron tomography reconstruction algorithm ICON with GPU.

2. Deng Y., Chen Y., Zhang Y., Wang S., Zhang F.* and Sun F.* (2016), ICON: 3D reconstruction with 'missing-information' restoration in biological electron tomography. ***Journal of Structural Biology*** 195(1): 100-112**.** doi: 10.1016/j.jsb.2016.04.004.