EmSART – System requirements:

- Windows 7/8/10 or Linux with g++ version 4.8.4 (tested on Ubuntu 14.04)
- Nvidia CUDA 7.5 toolkit installed
- Multiple GPUs with a total memory size of the volume to reconstruct (16 or 32 bits per voxel) + some memory for the projection images (We use four GeForce Titan X with 12 GByte each).
- Enough RAM on host: There is no hard limit, but the more memory is available, the bigger the reconstruction can be.
- OpenMPI 1.6.5 installed for multi-GPU support

Start a reconstruction:

- For multi-GPU reconstruction with openMPI:
  - mpiexec -n #GPUsToUse EmSART -u configfile.cfg

- For single-GPU reconstruction without openMPI:
  - ./EmSART -u configfile.cfg

Configuration file settings:

Parameter	Possible values	Description
CudaDeviceID	For a default four GPU setup: 0 1 2 3	List of the device IDs to use.
ProjectionFile	MRC image stack or DM4 file series with index in filename: file_01234.dm4	Filename with the projection images. Note: EmSART limits the number of interpolation steps why the original unprocessed tilt series is used here. I.e. not the aligned stack as produced by IMOD.
OutVolumeFile	File name with ending .em, .mrc or .rec.	Filename of the reconstructed volume.
RecDimensions	x-dimension must be a multiple of 4, y-dimension must be even, z-dimension is unrestricted.	Dimensions of the volume to reconstruct in voxels. X/Y correspond to the projection image plane at 0° tilt, Z is the thickness. The Y-axis is the tilt axis.
VolumeShift	X Y Z value	Volume shift with units “pixels of the micrograph”; this makes the shift values independent of the voxel size/binning.
MarkerFile	EM file format	Filename of the markerfile containing tilt series alignment information. Format as used in “Clicker”.
Lambda	1	Relaxation factor used in SART. As we run only one iteration, Lambda has to be equal to 1.
Iterations	1	Number of SART iterations. Experiments showed, that best quality is achieved with Lambda = 1 and iterations = 1.
UseFixPsiAngle	false	determines if the markerfile contains in-plane-rotation angles for each projection image. If set to true, Psi-Angle must be set.

PsiAngle	Angle in degrees	In-plane-rotation image, if above is true, set the angle for all images
PhiAngle	0	beam declination (use with care, not well tested)
OverSampling	1-10 (more is possible but impact decreases)	Number of rays per pixel (for each dimension). See also the super-sampling SART paper. Also note that a voxel size > 1 corresponds to oversampling.
VoxelSize	1 = Same size as projection pixels; 2 = One Voxel has the size of two pixels, etc.	Voxel size in pixels (the base unit in EmSART is "1 pixel")
CtfMode	True/false	enable CTF correction, if true, CTF file must be provided
CTFFile	Filename of the EM-file containing the defocus values	Defocus values for each projection.
CTFBetaFac	360 0 0.01 0	CTF correction parameters: first number gives the first maximum of the Thonrings in the power spectrum in pixels. No CTF correction is performed for frequencies below that value. The other three parameters (a, b and c) define the dampening envelope, where $\mathrm{env} = e^{-ax-bx^2-cx^3}$, with env the envelope function, a, b, c, the three coefficients and x the spatial frequency in 1/nm.
Cs	e.g. 2.7	Cs value of the projection lens, needed for CTF correction (in mm), can be omitted if CtfMode = false.
Voltage	e.g. 300	Acceleration voltage of the microscope in kV, needed for CTF correction, can be omitted if CtfMode = false
SkipFilter	True/false	Perform bandpass filtering on the projections or not before reconstruction
fourFilterLP	e.g. 1000	bandpass filter parameters in pixel: Low pass
fourFilterLPS	250	Low pass sigma
fourFilterHP	0	High pass
fourFilterHPS	0	High pass sigma
SIRTCount	1:15 (a number equal to the total number of projections would turn SART equal to SIRT)	Contrast = ordered subset size in super-sampling SART paper (1=perfect for sub-tomo-avg, 10-15 = good contrast)
BadPixelValue	X - Depends on micrograph pixel values	All pixels above X-times the mean value will be filtered out to cancel out X-rays
CorrectBadPixels	True/false	Correct for bad pixels or not
AddTiltAngle	Angle in degrees	Rotate the volume along the tilt/Y-axis by X degrees (not rad)
AddTiltXAngle	Angle in degrees	Rotate the volume along the X-axis by X degrees (not rad)

FP16Volume	True/false	Use 16Bit floats instead of 32bit to save memory
WriteVolumeAsFP16	True/false	If the one before is true, the result can be converted to 32bits before writing to disk
ProjectionScaleFactor	1 or 1000	Scale the images by 1000 (that's a good value...) if you use 16 bits. If 32 bit floats, use 1
ProjectionNormalization	“std” or “mean”	normalize the projection images by standard deviation or alternatively by the mean value to normalize the tilt series: $$image = \frac{(image - mean)}{std}$$ or $$image = \frac{(image - mean)}{mean}$$
WBP	True/false	false -> use SART not WBP. If set to "true", one can also use standard WBP with a ramp filter.

Tilt series alignment:

The marker file used to store the tilt series alignment information is similar to older TOM-toolbox markerfiles. Its structure is as follows:

- One column for each projection image in the tilt series
- 10 lines, where:
  1. Tilt angle in degrees
  2. X-position in pixel of marker in micrograph (a negative value indicates to skip the projection)
  3. Y-position in pixel of marker in micrograph (a negative value indicates to skip the projection)
  4. Not used
  5. X-shift in pixels
  6. Y-shift in pixels
  7. Not used
  8. Not used
  9. Magnification change (image scaling factor)
  10. In plane image rotation angle
- Third dimension is used for each clicked marker, but only the alignment information of the first marker is used.

The markerfile can be created either using the tilt series inspection and alignment tool “Clicker” or by converting an IMOD alignment using the following MATLAB script:

The matlab script will convert an IMOD aligment to a pseudo-marker file that can be used with EmSART. Just save the matrix as an EM-file:

```
m = ConvImod2EM('filenamebase');

emwrite(m, 'markerfile.em');
```

CTF correction:

The defocus values are given in a two-dimensional matrix saved in EM-file format. One row per projection, five columns where:

1. CC-value (not used in EmSART)
2. Lower defocus value
3. Upper defocus value (for astigmatism)
4. Astigmatism difference (equals column 3 minus 2)
5. Astigmatism angle in radians

Other CTF parameters are given in the reconstruction configuration file, such as Cs, acceleration voltage and dampening envelope.