

Quick manual to 'FIB2CAM'

Interpretation and interpolation of G-code for stream file generation for FIB/SEM instruments

Version 0.1

Frank Niessen

E-mail: contact@fniessen.com

03/2019

1 CAM2FIB

CAM2FIB is a program for interpretation and interpolation of G-code to create stream files for FIB/SEM instruments. The currently implemented output format is designed for use with an FEI Helios NanoLab G3 instrument, but may easily be adapted to other instruments. This code accepts most G-code files as an input and was tested with the G-code dialects *RS274D*, *WinCNC* and *FreeCAD*. In its current implementation the overall patterning time is equally distributed over all patterning points. Machining or printing of 3D-structures is therefore currently limited to layered structures. This manual gives an introduction to the application of the program. For further input on the build-up and functionality of the program please read the associated research paper.

This program is provided free-of-charge with an MIT-license. If you find this program useful and you find application for it in your research I would be grateful for a citation to the associated paper:

2 SYSTEM REQUIREMENTS

The program was developed in MATLAB and requires a valid MATLAB installation for execution. The functionality of the program was tested for MATLAB 2016 b and later versions. The installation of the MATLAB statistics toolbox is not essential to the functionality of the program, but is required to plot density maps for evaluating the interpolation of the beam path.

3 EXECUTING *CAM2FIB.M*

All folders and functions need to be added to the MATLAB path before executing the program. The program is executing by running file *CAM2FIB.m*. The default folder for input files is *Input_Gfiles* in the main directory. Stream files and plots are saved in the folder *Output_StreamFiles* in subfolders named after the input G-code file.

4 PARAMETER SETTINGS

The section *%% Initialization* in *CAM2FIB.m* contains parameters that can be adapted for different applications of the program. The parameters, their purpose and potential default values are listed in the following table.

Name	Purpose	Description	Datatype	Allowed values	Default value
File settings					
<code>f.unitLabel</code>	Output length unit	Length unit of the machined structure	String	m, mm, μ m, nm, Å	nm
<code>f.overlap</code>	Relative overlap	Relative overlap of beam path $(1 - \frac{BeamPitch}{BeamWidth})\%$	Double	<i>-inf</i> – 100	70
<code>f.dcrit</code>	Min. G0 distance	Minimal segment length for beam blanking	Double	0 – <i>inf</i>	0

f.binFac	Interpolation fac.	Interpolation of entire beam path binning factor: <1: No interpolation >=1: Interpolation with indicated binning factor	Double	0 – inf	1
Visual output options					
Out.scrPrint	Flag Screen Print	Screen output of G-code interpreter [on/off]	Boolean	0, 1	0
Out.plot	Flag Plotting	Plotting of maps [on/off]	Boolean	0, 1	1
Out.scatter	Flag Scatter Plot	Scatter instead of Line plot of beam path [on/off]	Boolean	0, 1	1
Out.plotCnt	Plot Count	Number of plot updates during G-code interpretation	Double	1 – inf	100
Stream file output settings					
str.DAC	DAC type	Type of DAC (currently only tested with type 's16')	String	s16	s16
str.nrIter	Nr. Of iterations	Number of pattern iterations	Integer	1 - ?	1
str.tMach	Pattern machining times [s]	Array of overall pattern machining times [s] (Keep empty for no file output)	Double array	1 - ?	60
str.fType	Outp. File extension	File extension of stream file	String	/	str
Scaling parameters					
PIA.rot	Pattern rotation [°]	Clockwise pattern rotation [°]	Double	/	0
PIA.flip	Mirroring operation	Mirroring on horizontal and vertical axes [on/off]	Boolean array	0, 1	[0 0]
PIA.scalFac	Linear scaling factor	Linear scaling of pattern	Double	0< x < inf	1
PIA.cal	DAC calibration	DAC calibration factor $(\frac{Bits}{meter*Mag})$	Double	0< x < inf	.3161e6
PIA.max	DAC resolution	DAC resolution in X and Y	Integer array	0< x < inf	[65536 56576]
PIA.offS	Offset pattern [bit]	Offset of pattern in X and Y	Integer array	/	[0 0]
PIA.relScale	Rel. pattern scale	Relative scale of pattern on DAC system [1] (Used to determine the magnification)	Double	0< x <= 1	.5
PIA.centrePattern	Flag Centre pattern	Centre pattern on DAC grid	Boolean	0, 1	1
PIA.availMagns	Mag. Selection	List of allowed standard magnifications	Double array	0< x < inf	[.1e3 .5e3 1e3 2e3 5e3 10e3 20e3 30e3 40e3 50e3 60e3]

5 PLOTS

While G-code with multiple z-layers can be interpreted and interpolated, the graphical output of the program is optimized towards processing of single layer beam paths. In the case of multi-layer input maps may overwrite lower z-layers, leading to a false representation.

Plots are saved as *tiff* files alongside with the stream files in a subfolder with the name of the input G-code file, which can be found in the folder *Output_StreamFiles* in the parent directory.

5.1 LIVE PLOT

The plot *LivePlot* shows the progress of G-code interpolation with an update rate specified by variable *Out.plotCnt*. The plot shows the beam path before application of any transformations.

5.2 MACHINING STRATEGY

The contour plot *MachiningStrategy* shows the beam path after application of any transformation. The colour coding indicates the order of patterned points. White broken lines indicate repositioning of the blanked beam.

5.3 DENSITY MAPS BEFORE AND AFTER INTERPOLATION

Density maps are 3D histograms that show the density of patterning points at different positions of the pattern. The overall pattern area is divided into equisized bins and the number of points in each bin is represented on a color scale. The plot intends to highlight areas of especially high and low point density. The map is plotted before and after interpolation of the entire beam path to visualize the effect of the interpolation operation.

5.4 SCALED AND TRANSFORMED PATTERN

The plot *ScaledAndTransformedPattern* shows the beam path after application of all transformations and scaling to the DAC coordinate system.

6 STREAM FILES

The stream files are written into a subfolder with the name of the input G-code file, which can be found in the folder *Output_StreamFiles* in the parent directory. The stream file can be directly transferred to the microscope computer and opened from the patterning control panel.

7 TROUBLE SHOOTING

If any unforeseen problems occur with the program please contact me via contact@fniessen.com.