urpec

URpec is a proximity-effect correction software designed for NPGS. In the future, it will be extended to other platforms. Urpec Generates a proximity-effect-corrected pattern file for electron beam lithography.

The input pattern file can be either a .dxf or .mat file. The output file has different colors, each of which receive a different dose. This function assumes that one unit in the input pattern file is one micron.

Urpec finds all of the polygons in your pattern. It then creates a 2D grid spanning your pattern and deconvolves the point spread function to achieve a uniform dose over your pattern. The deconvolved pattern is then used to calculate the dose modulation across the pattern. The deconvolution uses the a version of the CLEAN algorithm. Once the dose modulation is calculated, urpec optionally fractures your pattern into smaller polygons with uniform doses. If non convex polygons are used, urpec triangulates these polygons before fracturing.

Urpec outputs pattern files, a list of doses in a text file, and a \_fields.mat file. The \_fields file is used to create the run file.

Right now this is intended for use with NPGS. The script run\_urpec shows how to create a run file.

Urpec takes a single optional argument, config. config is an optional struct with the following optional fields:

1. dx: spacing in units of microns for the deconvolution. The default is 0.01 microns. It is also best to have the step size several times larger than the center-center or line-line spacing in npgs.
2. maxIter: maximum number of iterations for the deconvolution. Default is 6.
3. dvals: doses corresponding to the layers in the output file. Default is from 1 to 2 in 15 steps in units of the dose to clear.
4. targetPoints: approximate number of points for the simulation. Default is 50e6.
5. autoRes: enables auto adjustment of the resolution for ~10min computation time.
6. file: datafile for processing. This can either be a .dxf file or a .mat file. If it is a .mat file, the contents of the file should be a struct called polygons. The polygons struct should have at least these fields:
   1. p: a cell array of polygons. Each element of the cell array should be a nx2 array of coordinates describing the poylgon.
   2. layer: an array of numbers specifying the layer of each polygon according to the convention described above.

The layer scheme is as follows. The names for all layers should be numbers. Layers 1 and 2 of the input file will both be output to layer 1 of the output file. Layer 1 will not be fractured, and layer 2 will be fractured. Layers 3 and 4 of the input file will be output to layer 2 of the output filed, etc. If the polygons are not fractured, they are written with an average dose.

1. psfFile: point-spread function file
2. fracNum: maximum number of times to fracutre a shape
3. fracSize: minimum size for fractured shapes, in units of dx.

To run urpec, make sure it and its subfolders are on your path, and type urpec, and press Enter. To call this function with arguments, run urpec(struct('dx',0.005, 'subfieldSize',20,'maxIter',6,'dvals',[1:.2:2.4])) for example.

By: Adina Ripin [aripin@u.rochester.edu](mailto:aripin@u.rochester.edu), Elliot Connors [econnors@ur.rochester.edu](mailto:econnors@ur.rochester.edu), John Nichol jnich10@ur.rochester.edu

Version history

v2: handles different write fields and writes directly to dc2 format.

v3: writes all doses to the same layer but with different colors, PSF improvements, Entirely new fracturing algorithm