CCD\_to\_qspace: MATLAB package (Kiyo Akabori, 10/10/2014)

This document describes analysis of wide angle X-ray scattering using MATLAB. /example/waxs.m is an example script file: click on it to see it and modify it for different jobs. This script, when run from MATLAB, produces the following files: column.dat, matrix.dat, qr.dat, qz.dat, ripple\_10deg.dat, ripple.pdf, ripple\_qr.dat, ripple\_qz.dat. The pdf file is an image file. The dat files are ASCII files that can be imported to OriginPro for graphing, as described later.

To run a script, start MATLAB by typing “matlab” without quotation marks in a terminal in any directory; wait a bit. Once MATLAB is started, cd into “example” directory by typing

>> cd /home/kiyo/MATLAB\_UserFunctions/Functions/CCD\_to\_qspace/example

Here, “>>” means a command prompt in MATLAB. Once you are in “example” directory, enter

>> waxs

which will run “waxs.m” script. Make sure that the above mentioned files are produced after the script is finished. The script will also create some figures.

Let's go through the script in detail. Some commands are native MATLAB commands, so you can look them up online or in help.

To use CCD\_to\_qspace package, we need to tell MATLAB where relevant files are located. The path to the package is added like

>> addpath('/home/kiyo/MATLAB\_UserFunctions/Functions/CCD\_to\_qspace');

We also need to add the path to some necessary commands:

>> addpath('/home/kiyo/MATLAB\_UserFunctions/Functions');

>> addpath(genpath('/home/kiyo/MATLAB\_UserFunctions/Functions/Downloaded'));

These three lines are always necessary to use the package. Of course, if you move any directory, you have to modify these paths. Data files are added as

>> addpath(genpath('/home/kiyo/data/chess11'));

For your own analysis, this path (inside ' ') must be modified accordingly. For show command to work, enter

>> global MaskD

>> MaskD = uint8(ones(1024, 1024));

We next enter experimental parameters: wavelength (Angstrom), pixel size (mm per pixel), and S-distance (mm). This is done as

>> wavelength = 1.176;

>> pixelSize = 0.07113;

>> Sdist = 169.8;

Things on the left of “=” are variable names. You are free to name them whatever you like. They are simply inputs to the transform\_ccd2q function.

Let's load sample and background images. Type

>> a = load\_chess('ripple\_060\_cz.tif');

>> b = load\_chess('ripple\_061\_cz.tif');

Here, a is a sample and b is a background image. Take a look at these images by typing

>> figure; show(a, [0 2000]);

>> figure; show(b, [0 2000]);

[0 2000] is a gray scale value. Intensity < 0 is black and > 2000 is white. We need to find the beam position. For this, we plot qr and qz swaths like

>> figure; qzplot(a, []);

>> figure; qrplot(a, []);

Save the beam position in variables via

>> beamX = 32;

>> beamZ = 110;

Subtract the background by

>> img = a – b;

img contains the background subtracted data.

Then, we rotate the image about the beam (0.97 degrees in this case),

>> img = rotateAround(img, beamZ, beamX, 0.97, 'bicubic');

Now, we prepare some variables for the transformation. First, decide what q range you want to convert,

>> qr\_range = [0 1.8];

>> qz\_range = [0 1.0];

We also need q resolution, which is estimated approximately by

Δq ≈ 4π Δθ / λ ≈ 4π pixelSize / (λ Sdist) (1)

>> delta\_q = 0.0024

Set the angle of incidence (positive for nGIWAXS and negative for tWAXS),

>> omega = 0.2

We are ready to transform the CCD image to q-space. Type

>> q\_img = transform\_ccd2q(img, qr\_range, qz\_range, delta\_q, omega, beamX, beamZ, Sdist, wavelength, pixelSize);

Save the q-space image by

>> save\_q(q\_img);

This command creates files for the matrix, qr values and qz values. “column.dat” can be imported to OriginPro as x, y, and z columns. Use them to make a contour plot.

To plot a 2D image, type

>> figure; qshow(q\_img, [0 1000]);

You can save the image in various formats. To save the image within a script, see “waxs.m” script file.

To plot qz and qr swaths, type

>> [qz, Int] = qzplot\_q(q\_img, [1.483 1.5]);

>> [qr, Int] = qrplot\_q(q\_img, [0.19 0.21]);

To make a radial plot (from 1 to 2 A^-1, averaging between 5 to 15 degrees)

>> [q, Int] = radial\_q(q\_img, [1 2], [5 15]);

To make an annular (ring) plot (from 0 to 70 degrees, averaging between 1.3 and 1.6 inverse Angstrom)

>> [phi, Int] = ring\_q(q\_img, [0 70], [1.3 1.6]);

Use dlmwrite command to save those plots in an ASCII file. See “waxs.m”.

To record your analysis, create a script file similar to “waxs.m”. This will allow you to reproduce all parameters, background subtraction, data manipulation, and graphs by simply entering

>> script\_filename

without the extension “.m”.