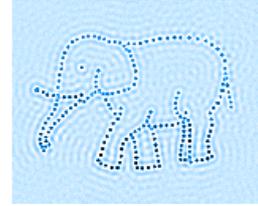


EMBO Practical Course on Image Processing for Cryo EM 1 – 11 September 2015



Practical 1: Introduction to basics of image processing with EMAN2

Log into a PC with the username and password provided and then on to Nomachine and start a terminal window according to the instruction sheet for each user. If you are not sure what directory you are in you can always type “`pwd`” and this will display the directory you are currently in.

Move into your prac-1 directory (`cd prac-1`).

In prac-1, the files `circle.hdf`, `rectangle.hdf`, `hhrot.hdf`, `micrograph1.mrc` and `micrograph2.mrc` are the ones you will use. Check by listing them (`ls`). (Reminder: basic UNIX commands are listed on page 3 of the Getting Started document).

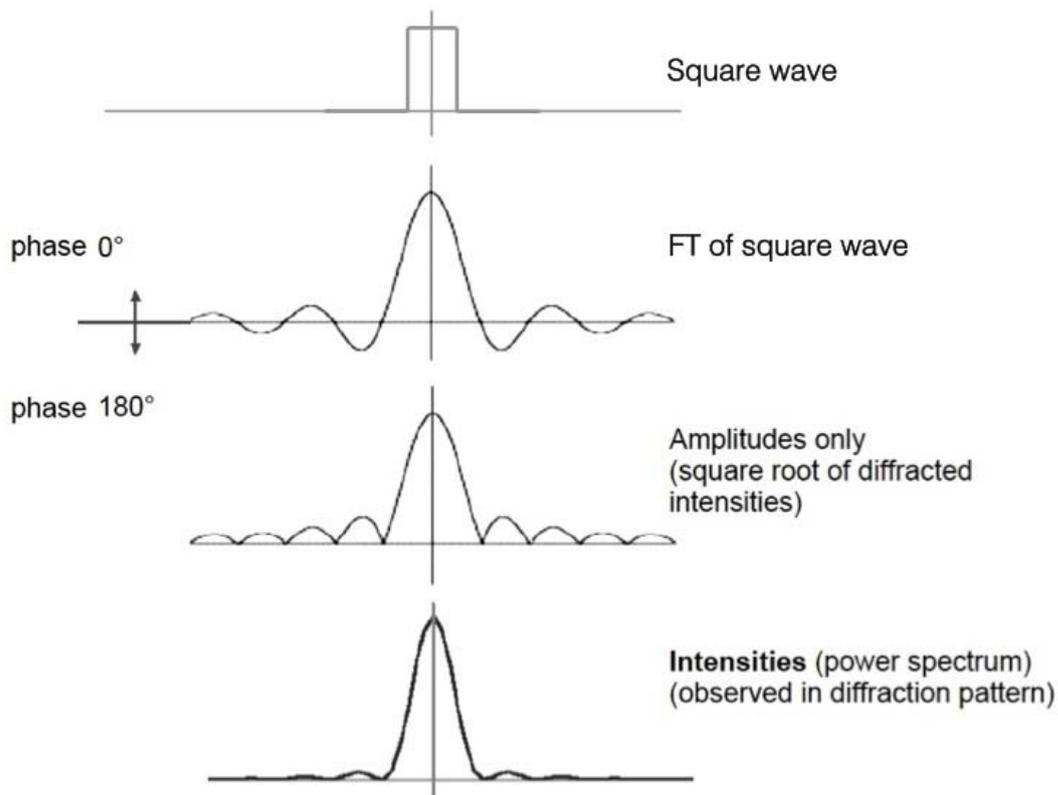
The purpose of this practical is to introduce you to images, their transform amplitudes and phases, and the effects of filtering. Then you will do manual defocus determination from micrographs, and finally some particle picking. You will use EMAN2.1, for which there is extensive, although not complete, documentation:

<http://blake.bcm.edu/emanwiki/EMAN2>

Part 1: Images, Fourier transforms and filtering

The purpose of this introductory part is to show you how to look the Fourier transforms (FTs) of images. FTs can't be displayed like simple images, since they contain two components at every pixel, an amplitude and a phase angle. The amplitudes can be represented as grey scale values, as normally used to represent the transform of a micrograph, in order to check for drift and determine defocus. The phases are not normally viewed, but can be represented as colours on a cyclic scale so that 0 and 360° are the same colour. If you want to understand image processing, it is important to understand FTs and diffraction. FTs present the image as a function of spatial frequency (1/distance) instead of distance. Since many operations need to be done on specified ranges of spatial frequency, they are done in the frequency domain. Here you will view two simple 2D images and their FTs.

In one dimension, the object and its FT can be plotted as graphs:



To display images and their FTs, start by typing the command:

e2filtertool.py circle.hdf

This will give you a display window showing a white disk and a control window with drop down function menus. You can center the image in the display window by dragging with the right mouse button or by resizing the window.

Images, amplitudes and phases

Click on the image with the middle mouse button to bring up an image control window. It has options to display the real image, its Fourier transform (*FFT*) shown as amplitudes with colour-coded phases, or the amplitudes (*Amp*) or phases (*Pha*) separately. Look at the amplitudes. You may need to choose *auto contrast* and increase the view magnification (*Mag* slider or mouse wheel). You should see a set of concentric rings around a central maximum, a 2D version of the graph above, similar to the Thon rings that are used to judge the quality of micrographs.

Then try *FFT*. If you magnify the display more, you will see that the phases are alternating between red and green (0 and 180°) in adjacent pixels. This is because the phase origin is at the center of the image, to make the real image display correctly. To see how the phases vary with the phase origin in the bottom corner, choose *Xform* from the function menu on the e2filtertool window and choose *phaseorigin.tocorner* on the right menu. Then check the box to the left of the *xform* menu bar. The FFT should

now show alternating rings of phase reversal, like the FT of the square wave. Returning to *Real* on the image display will show you how this affects the image appearance. You may need to reduce the image display magnification to see something. To restore the image, uncheck the tick box at the *xform* menu. Repeat these operations with the image **rectangle.hdf**. To see the transform amplitudes, you may need to reduce the *max* value after choosing *autocontrast*. Note the inverse relationship between the repeating bands in the FT and the dimensions of the rectangle.

Filtering

Close the previous windows and display the circle image with **e2filtertool.py circle.hdf**.

Choose *filter* from the dropdown menu and check the box to the left of it. The right hand drop down menu will contain a list of available filters. The ones you should try out are lowpass, highpass and bandpass filters using Gaussian (smooth) or tophat (sharp) fall off. Start with *lowpass.gauss*. If you hover the mouse over the titles or parameter boxes of the different sliders, while the window is selected, you will see a short text explanation of what they do. The best one to use is *cutoff_abs*, which gives the filter radius as a fraction of the Nyquist frequency, which is 0.5 at the maximum radius. Move the slider over the range, allowing some time for the graphics window to refresh.

What does the low pass filter do to the image?

Bring up the image control window by clicking on the image with the middle button. Using display as *real* or *Amp*, look at the effects of filtering on both the image and the transform. Can you explain the relationship between the changes in the image and the effects of the filter on the FT? Repeat with high pass and band pass, and then try the series with the tophat filters.

Now close the filtertool window and start a new one with

e2filtertool.py rectangle.hdf

With this image, you can again look at the transform amplitudes (if *autocontrast* doesn't give good results, try reducing the *max* slider on the image control window). Then look at the FFT and try out the various filters - what happens with high pass and band pass, and how do the effects differ if you use the Gaussian or tophat versions?

You may need to restart e2filtertool.py if you do too many different operations. And note that the graphics window may not refresh until the filter is reapplied.

When you have investigated the behaviour of images, FTs and filters with the circle and rectangle, look at the effects of filtering a slightly more realistic image.

e2filtertool.py hhrot.hdf

(depending on your system, you may need to use the unrotated version hh.hdf)

How are different features in the image affected by low pass and high pass filtering?

Now you can try the filters on a real micrograph:

e2filtertool.py micrograph1.mrc

In the filtertool control window, choose filter – lowpass.gauss, and set the pixel size (check *apix*) to 2.44 Å. Select *cutoff_freq* (and deselect *cutoff_abs* or any other options if selected). Now you can check the effects of filtering on the micrograph, which is of GroEL on a carbon film, recorded with a DE20 detector on a Polara microscope. Try the other Gaussian filters to see which makes the particles easiest to discern. Now try micrograph2.mrc – this one was taken closer to focus and it is harder to see the particles.

Part 2 Defocus measurement

In this part you will use EMAN2 to manually measure the defocus of the 2 micrographs. This is usually automated with a script but doing it manually will show you how the matching of parameters is done and how the parameter choices affect the outcome.

e2evalimage.py micrograph1.mrc

This brings up 4 windows, though some may be covered by others. Find the control window, the plot window, image and FFT windows. Set the 4 data parameters at the bottom of the control window: *A/pix* 2.44, *Voltage* 300, *Cs* 2.3 and *Box size* 256.

You will probably need to adjust the scaling on the various displays. For each display window, middle click will bring up its own display control panel.

Starting with the plot window, use its control panel to set *Y Min* to -2 and *Y Max* to 10 (entries near the bottom of the control panel). This will put the graph on a better scale.

You can then close this control window. On the evalimage control panel, choose *refit*. On the plot, the blue curve is the fitted CTF and the black curve is the radially averaged FT of the micrograph. The radial curve is fine if there is no astigmatism in your image, but in general that is not perfect, so you also need to check the 2D presentation of the fit, on the FFT window. Resize the window or use the right mouse button to center the diffraction pattern, and bring up its control panel (middle click) to increase the mag and optimise the contrast. The fitted zeros are marked by circles – these would be elliptical to match astigmatism.

The micrograph window shows the size of the tiling boxes used to calculate the FT – the results of all boxes are averaged to give a better signal:noise ratio on the resulting curve. You can see how this changes on the graph by trying 512 (evalimage control panel box size), instead of 256.

Now examine the effects of the other key parameters (evalimage control window) on the outcome. Select refit each time you do an adjustment and see if the fit improves or gets worse.

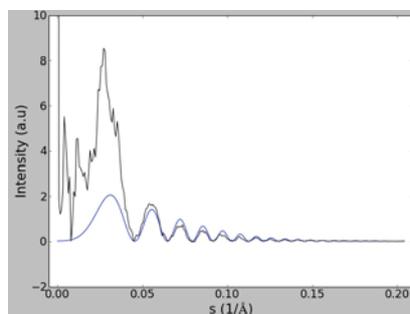
B factor

% Amplitude contrast (normally 7%)

The main problem is that there are strong features in the first part of the data (black plot) before the first zero, which are not fitted by the CTF model. Can you explain what these arise from? You can estimate what spacings the 2 or 3 sharp peaks correspond to, by estimating their position in $1/\text{Å}$ units, and then take the reciprocal to give the spacing in Å. (hint – you can measure features on the micrograph with the measure

(*Meas*) tab on the image control window (middle click)). You will need to set 2.44 Å /pixel and magnify the display so that you can see the measurement line, which is black. Use the mouse to draw a line between the features you want to measure. The result will be shown as length in Å (*Len*).

Example of a fitted CTF:



When done, close all the display windows and repeat the defocus determination for **micrograph2.mrc**.

Part 3 Particle picking and initial 2D classes

The purpose of this section is to introduce you to manual and automatic particle picking. If you can get the automated picking to work well enough, you can then do an initial alignment and classification of the picked particles. There is (currently fairly minimal) documentation on the automated picking program:

<http://blake.bcm.edu/emanwiki/EMAN2/Programs/e2boxer>

e2boxer.py micrograph1.mrc

This will open a control window, the micrograph display and a particle window. The automated boxing is called *swarm*. Set the particle diameter to 60.

Enlarge the micrograph display (mouse wheel or use image control panel) to make the particles clearly visible. To start the automated boxing (Boxing tool *swarm*), start by manually picking a few clear and ideally, isolated, particles, both side and end views. Center them with the mouse. The program will start to add automatically picked particles, and this will update as you make more selections and adjustments. You can try to improve the selection by trying the different options – *selective*, *more selective*, *threshold*, and using the *interactive threshold*, and *proximity threshold* to avoid overlapping or repeated particles. If you get stuck, start over by using *clear boxes*. The number of picked particles is shown at the bottom of the control window. It should be possible to get 200 or more particles from this micrograph, but to get good ones will take some trial and error. As you pick manually, the automatic selection changes. Check the quality of the autoboxing by scrolling up and down through the particles window.

When you are satisfied, choose *write output*. Click on *micrograph1* and OK. The stack of picked particles will be written in a particles subdirectory.

The alternative auto-boxing method, *gauss*, can also be used. To write out the picked particles, you need to select *swarm* and then *write output*. You will need to change the output file name, eg to `micrograph1_ptcls2.hdf`. Which method gives better results?

It will be much harder to pick from **micrograph2.mrc**, which is closer to focus. See how well you can do with the auto-boxing programs.

Initial class averages

With 150 or more particles, it is possible to do an initial alignment and classification, using the command `e2refine2d.py`. This command is documented on the EMAN2 wiki:

<http://blake.bcm.edu/emanwiki/EMAN2/Programs/e2refine2d>

It will take a few minutes to run. The stack of picked particles will be in your `particles` subdirectory:

cd particles

`e2refine2d.py --iter=10 --naliref=4 --ncls=4 --input=micrograph1_ptcls.hdf`

The results of this refinement will be written in a subdirectory of the type `r2d_01` and the final iteration of class averages will be in `allrefs_10.hdf`. To view them:

cd r2d_01

`e2display.py allrefs_10.hdf`

You can also try this with the alternative data set picked with the *gauss* option. If you managed to get a data set from `micrograph2`, you can also try aligning that one.

Sparx documentation on Gauss convolution based automatic particle picking

The micrograph is high-pass filtered using a Gaussian function with half-width derived from the particle window size. If requested, the overall average is subtracted and the micrograph is squared resulting in a *variance image*. For data with low contrast, this step significantly improves the detection rate. The micrograph is convolved with another Gaussian function with width derived from the expected particle size (can be adjusted using slider). Locations of putative particles are selected as maxima of the Gaussian-convolved image with additional exclusion of locations that are too close to each other, as determined by the output window size.

Extra section for advanced students

A whole data set of picked, CTF corrected and normalized GroEL particles is provided in the subdirectory **prac-1-extra**, along with the information file `InfoFile.txt`. If you are familiar with refinement and reconstruction procedures in EMAN2, you can try to get a good 3D reconstruction. Documentation on EMAN programs:

<http://blake.bcm.edu/emanwiki/EMAN2/Programs>