MAPFoSt Documentation

Elias Wang

8/12/16

Table of Contents

1. Introduction

This guide provides some documentation for my MAPFoSt algorithm, adapted from Jonas Binding's "Low-Dosage Maximum-*A-Posteriori* Focusing and Stigmation" paper.

Summary of algorithm:

First take multiple images at different known test aberration values (relative to the starting value), then feed this into a minimization function to find the most likely starting aberration value. Use this information to update WD/stig.

2. Using MAPFoSt

The complete auto-focusing algorithm is used by calling 'MAPFoSt.m'. This function encapsulates the acquisition of the test images, the aberration estimation, and the final adjustment of the working distance (WD) – it should replace 'CMD_AUTO_FOCUS_FINE'. The function has the following prototype:

```
[z,finalWD,I1,I2]=MAPFoSt(ImageHeightInPixels,ImageWidthInPixels,DwellTimeInMicroseconds,FileName,FOV,maxiter,fallback,verbosity)
```

Description of input and output parameters:

<u>INPUTS:</u>

ImageHeightInPixels, ImageWidthInPixels: desired dimension of test images
DwellTimeInMicroseconds: test image dwell time
FileName: temp filename for saved images
FOV: image Field of View
maxiter: max number of iterations if algorithm fails
fallback: boolean, use Zeiss autofocus after maxiter
verbosity: boolean, print statements

<u>OUTPUTS:</u>

z: (relative) aberration estimate, in um
finalWD: vector of final WD and stigmation values set by algorithm
I1,I2: two test images taken

Notes about MAPFoSt variables:

- maxiter, fallback, and verbosity are optional parameters with default values of 1, true, and false, respectively.

- Test aberrations (T1,T2) are hard-coded to +-15um, see section 3.2 for more detail.
- NA is hard coded, see section 3.1.
- sigma represents the estimated noise based on the two test images
- Kx, Ky are the wave vectors. Using mod instead of circshift seems to provide better back-compatibility.
- init is the initialization point for the defocus estimation.

Inside the MAPFoSt function, we maximize the posterior probability by minimizing -ln(posterior). This is done using minimize.m and MAP.m. The MAP function calculates the -ln(posterior) using a given aberration, FFT of two test images, known test aberrations, NA, sigma, Kx, and Ky. It returns both the value and the derivative of the -ln(posterior) at the given aberration.

Notes about MAP variables:

- A_max represents the maximum aberration when using a uniform prior.
- A_sigma represents the standard deviation when using a Gaussian prior.
- cutoffx,cutoffy are used for selecting which subset of wave vectors to use. 50% of the height/width represents 100% Nyquist.
- MTF represents the MTF model
- dMTFdz is the partial derivative of the MTF with respect to defocus
- f is the value of -ln(posterior), note we are summing across all wave vectors used
- df is the derivative of -ln(posterior)

3. Determining Hyper-parameters

There are several hyper-parameters in the MAPFoSt algorithm that were roughly tuned empirically.

3.1 Numerical Aperture (NA)

There are 3 MATLAB files written to help automate the process of empirically determining the correct value for NA. The process is essentially just looking for the NA value that minimizes the MSE of the MAPFoSt estimation for a given data set.

The first script to run is MAPFoSt_NA_acquire_images. This will sweep through defocus values from -25um to 25um while saving the two test images at each point, allowing for us to perform the MAPFoSt estimation post hoc.

The next script is MAPFoSt_NA_calculation. This performs 4 stages of grid search: 4 orders of magnitude logscale, 2 orders logscale, 0.4 orders linear scale, and finally 0.1 orders linscale. The idea behind this is to be more robust to the initial NA value chosen. For reference, the current value is 0.0079. The MAPFoSt_NA_helper function returns the MSE of the MAPFoSt estimation for a given NA value and data set (path).

3.2 Test Aberration

The optimal test aberration range given in the paper is 13-16um. Therefore, we decided to use 15um. A quick comparison to 7um seemed to confirm that 15um was a decent choice. However, further testing could be done.

3.3 Wave Vectors

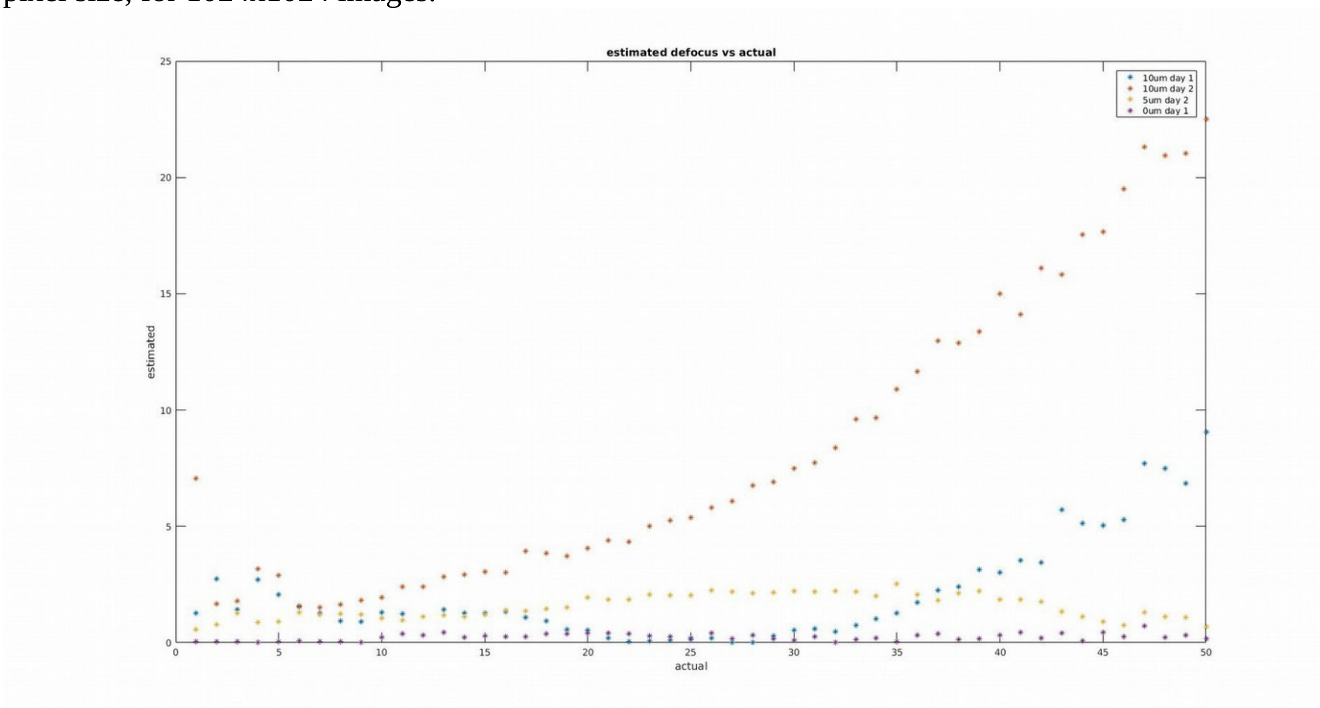
The paper performed some experiments testing what range of wave vectors to use, arriving at the result 25% Nyquist. In general, but not always, increasing the range of wave vectors used also increases the estimation accuracy and computation time. Therefore, the desired compromise between speed and accuracy may need to be tuned. Currently, we have found using 50% Nyquist and 128x128 image size or 25% Nyquist and 256x256 to be reasonable (see section 3.4 for more details about image size).

3.4 Image Size

Not surprisingly, we have found that increasing image size (pixels) generally seems to increase accuracy. However, larger images increase both imaging and computing time. We have found that, for our system, the optimal image size range seems to be 128x128 to 256x256.

3.5 FOV

The field of view that the test image are acquired at seems to affect estimation accuracy. However, we did not see a clear trend in our limited data. Below is a plot of estimation error versus pixel size, for 1024x1024 images.



In general, it seems that pixel sizes of 5-15nm seem to be the safest choice. This corresponds to 5.12-15.36um^2 FOV. Currently we use 8.192um.

4. Final Notes

The integration of MAPFoSt into WaferMapper 1.88 is done by adding another option to the MontageParametersGUI, IsAFOnEveryTileMAPFoSt. When selecting this option, all auto-focusing will be done with the MAPFoSt algorithm while auto-stigmation will be done with the built in Zeiss function. In addition to doing AF on every tile, this option will also perform an AFASAF once per montage.