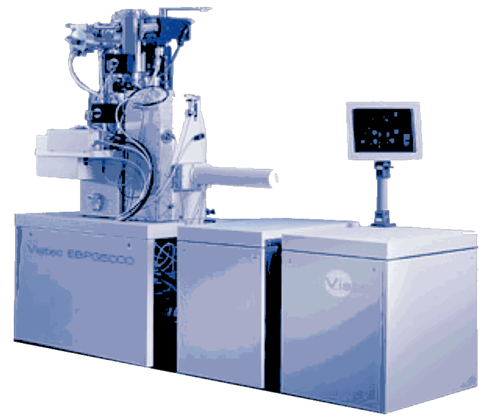
Analysis of electron-beam deflection noise with open-source software

Michael Rooks *Yale Institute for Nanoscience and Quantum Engineering*
Richard Tiberio *Geballe Laboratory for Advanced Materials, Stanford University*

We are not comparing vendors or machines in this presentation.

We used a variety of tools to test the programs, but we did not make any systematic comparisons.

For some of the exposures we purposely added noise to test the algorithm.



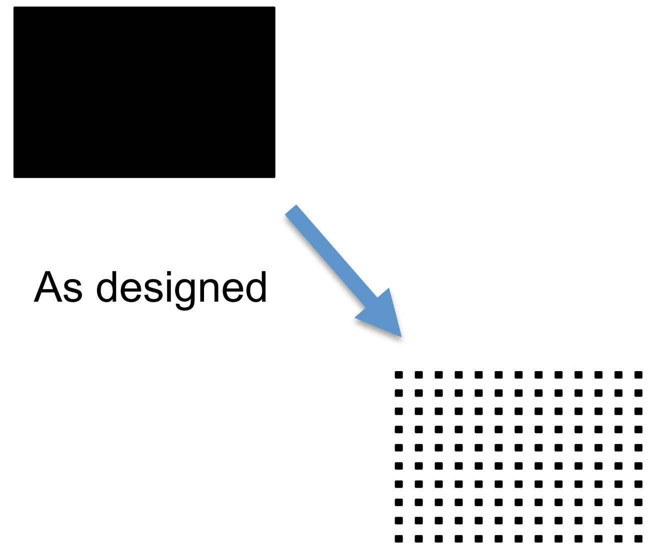
Problem: Measure quantitatively the deflection noise of an electron beam writer to diagnose noise problems, or to compare performance.

Line edge roughness (LER) is not an ideal way to measure deflection noise, since LER is a convolution of resist performance and exposure tool deflection noise.

Dots-on-the-fly, that is, exposing a solid rectangle with a large pixel step, is an easy way to visualize deflection noise, but quantifying the noise with commercial software is expensive.

Solution: Combine dots-on-the-fly with open-source software

Printing dots-on-the-fly:

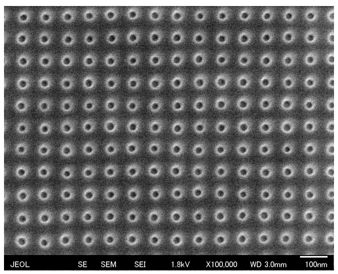


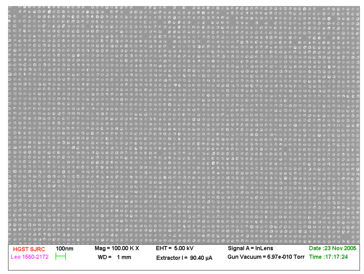
As designed

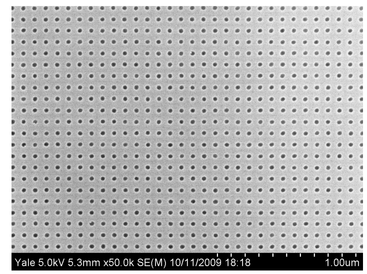
As written, underexposed with pixel step = 80 nm

Which type of image is the best for noise analysis?

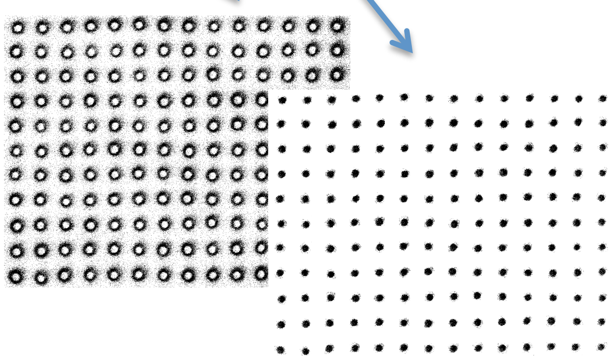
Resist	Metal	Etched holes
PMMA or ZEP	Cr, 10 nm	SiO_2 15 nm

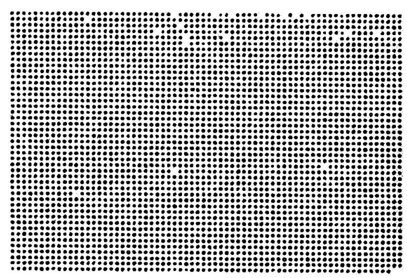


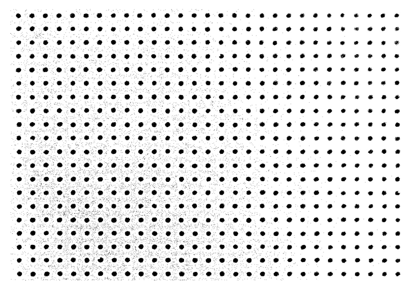




CHF_3 40 sccm
O_2 5 sccm
30 mTorr
100 W
3 min







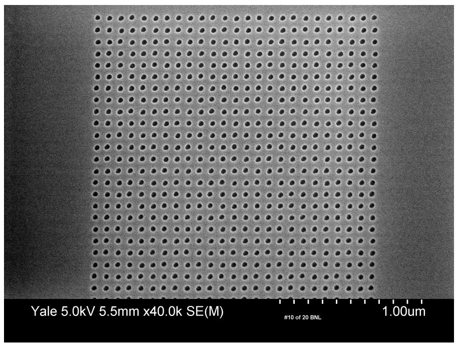
Process as doughnuts or blobs
Too few pixels, too much noise

Metal liftoff adds noise

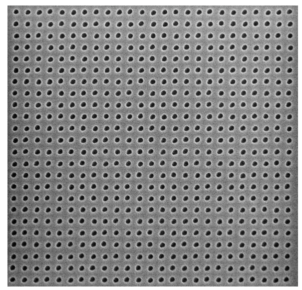
Etched holes are best

Image preparation

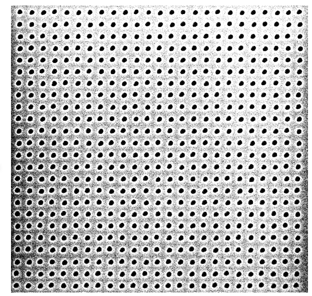
Manual image processing with an image editor such as Gimp or Photoshop:



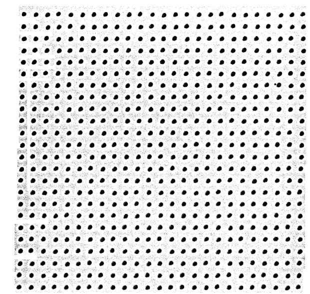
Original image



Cropped & rotated



Threshold filter



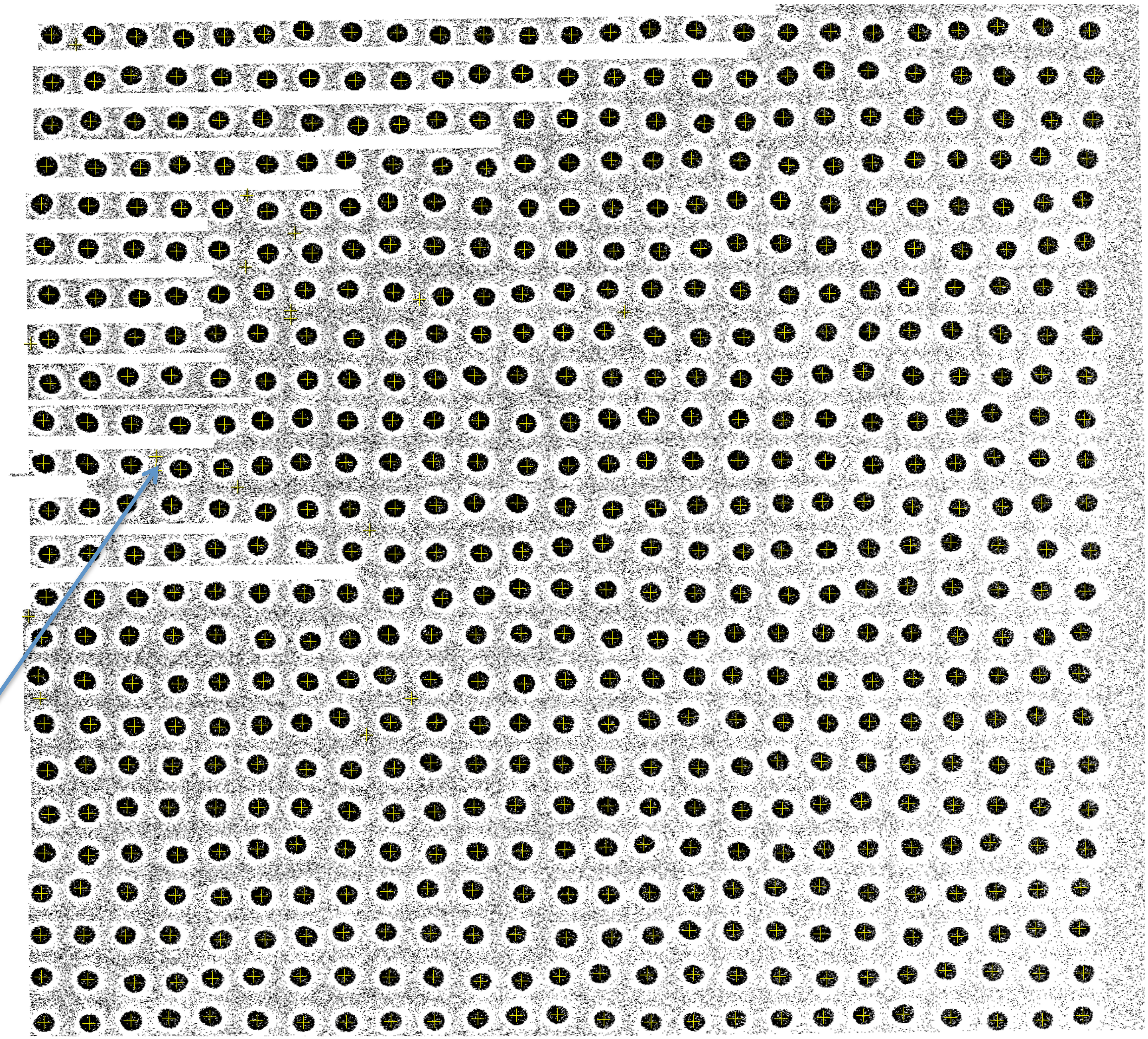
Manually erase dark regions from background

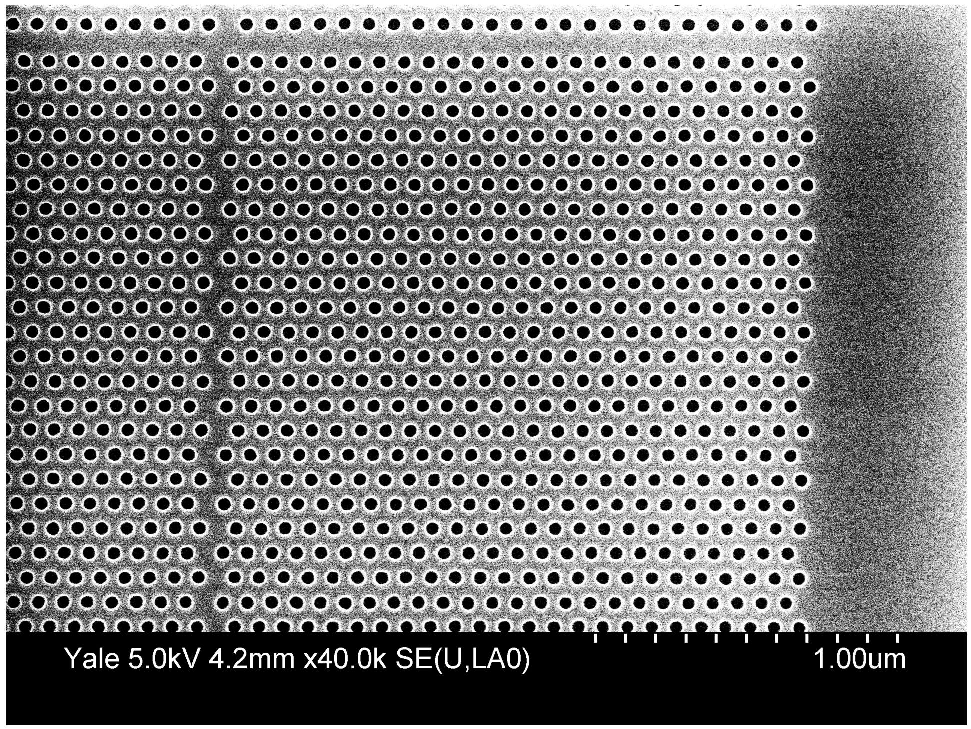
The program searches for dark regions. To avoid false hits, one may paint white regions between the dark dots. The user sets the minimum number of pixels in a dot, and this is called the

noise threshold

Regions with fewer contiguous dark pixels will be discarded as noise. Counting the number of dots is helpful when determining the optimal noise threshold.

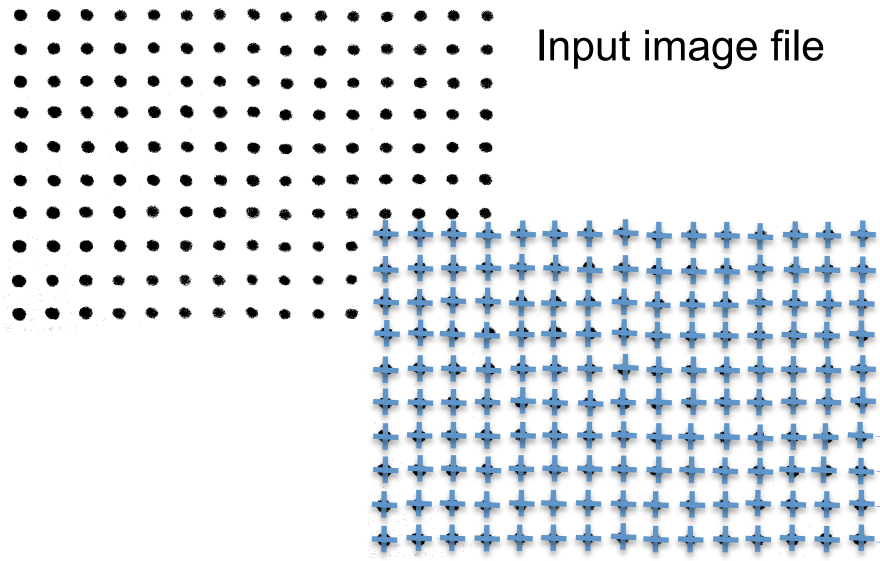
False hit

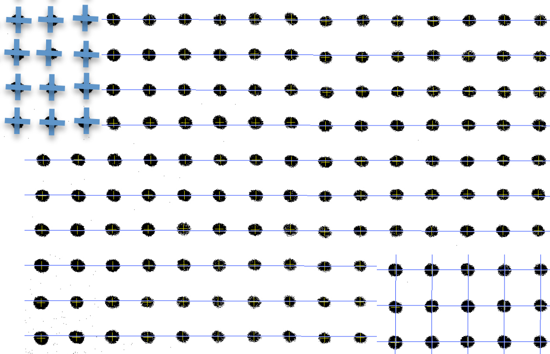


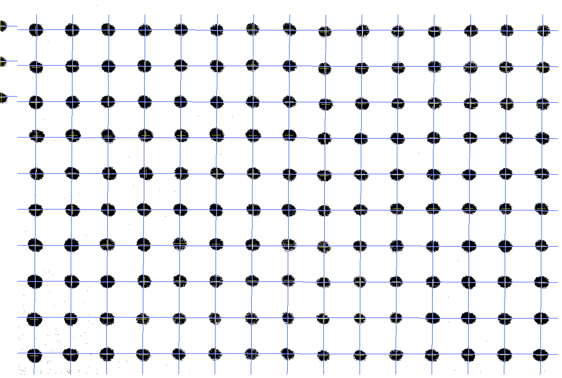


Hexagonal arrays of dots can be analyzed by using the option “--hex” of the program “dotnoise”.

Program execution







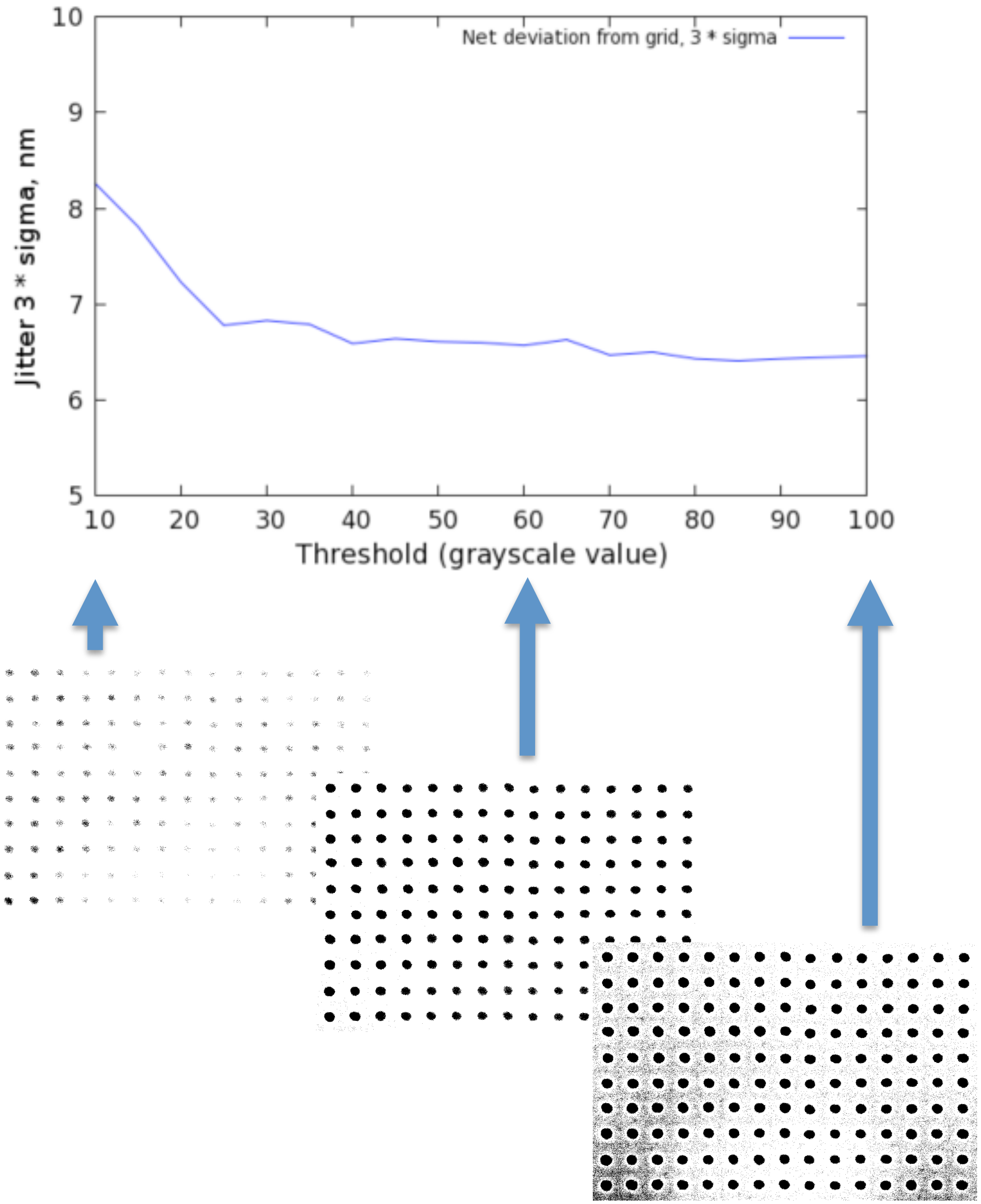
Horizontal and vertical gratings are fit independently, and my be non-orthogonal. This avoids problems with SEM drift during imaging.

The known pitch of the dots is used to calibrate the length scale.

Sensitivity to data preparation

Effect of the choice of threshold on the calculation of dot jitter.

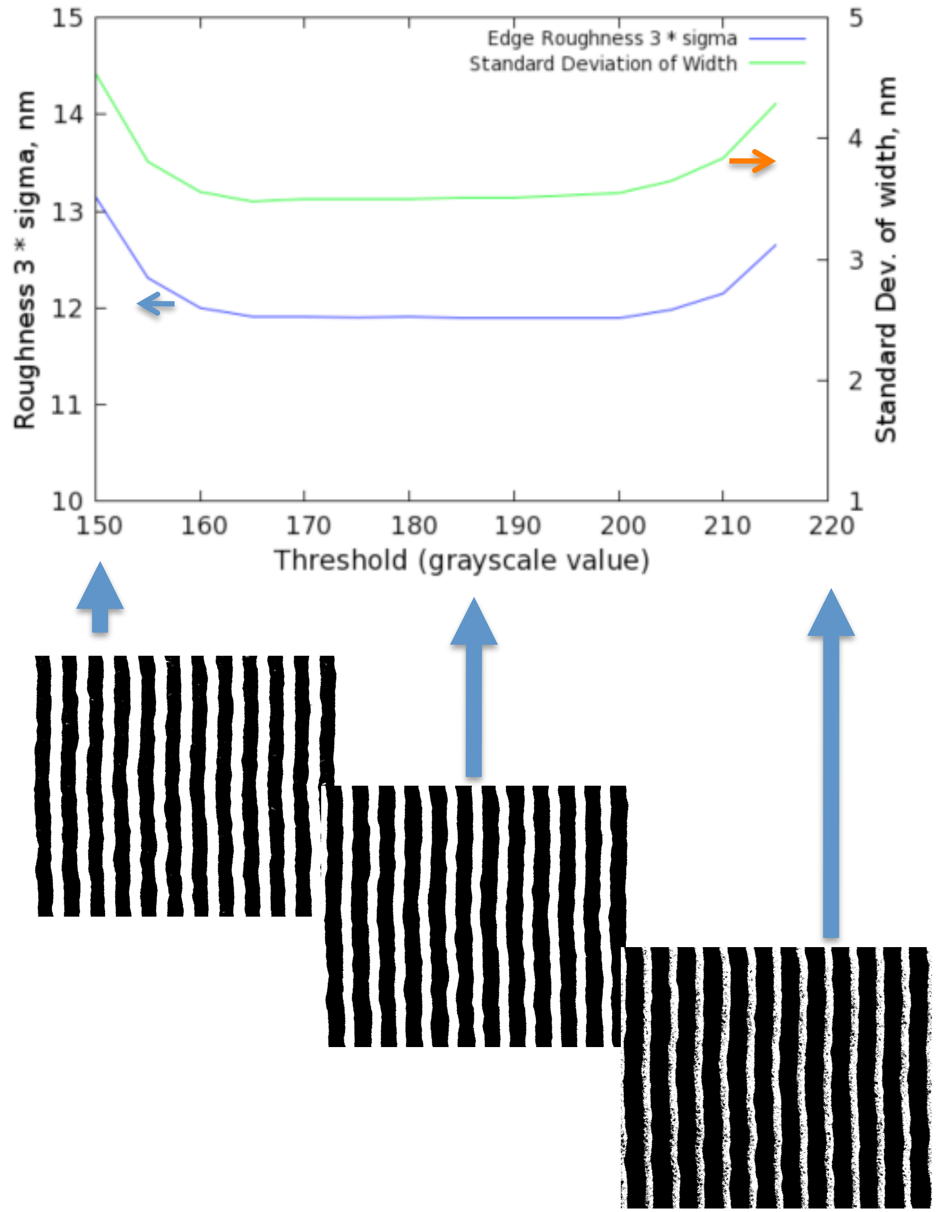
Threshold values over a broad range result in < 10% change in jitter



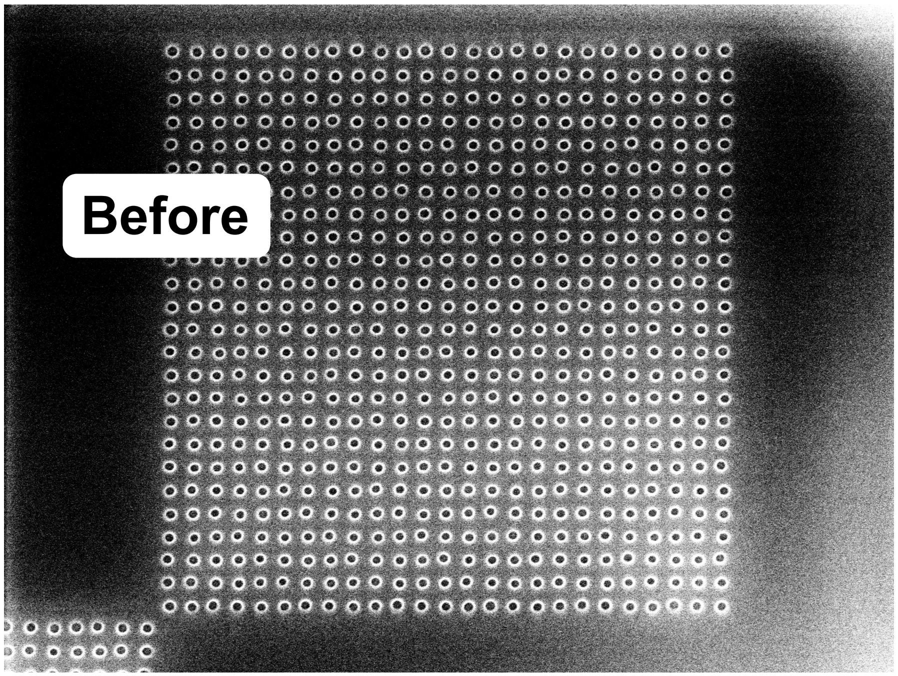
Sensitivity to data preparation

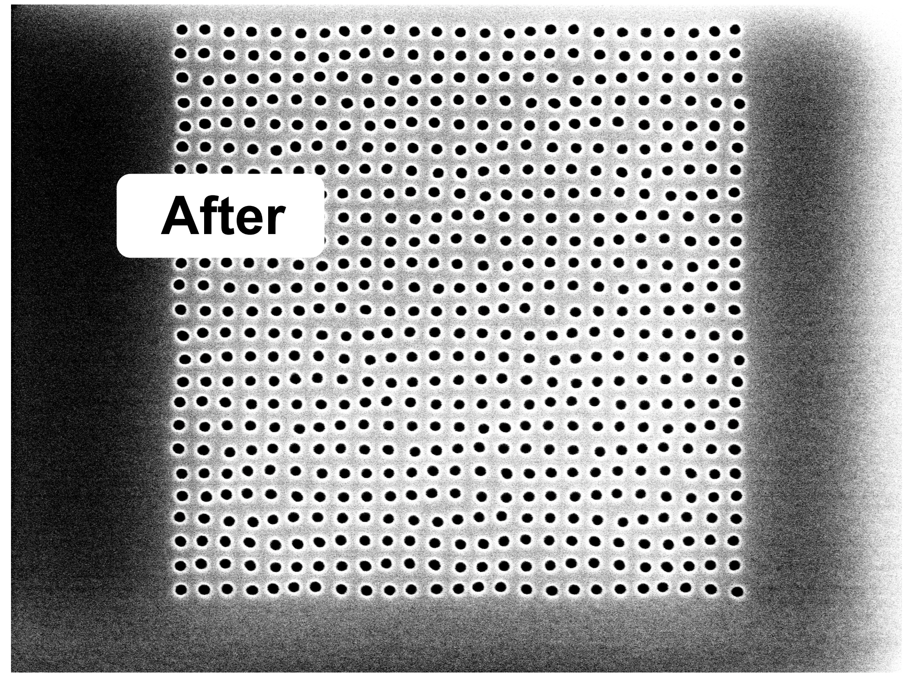
Effect of the choice of threshold on the calculation of line edge roughness and line width.

Threshold values over a broad range result in < 10% change in LER and width.



Purposely add a 20 kHz perturbation using a coil around the column:





3σ deviation from grid = 2.8 nm

3σ deviation from grid = 11.3 nm

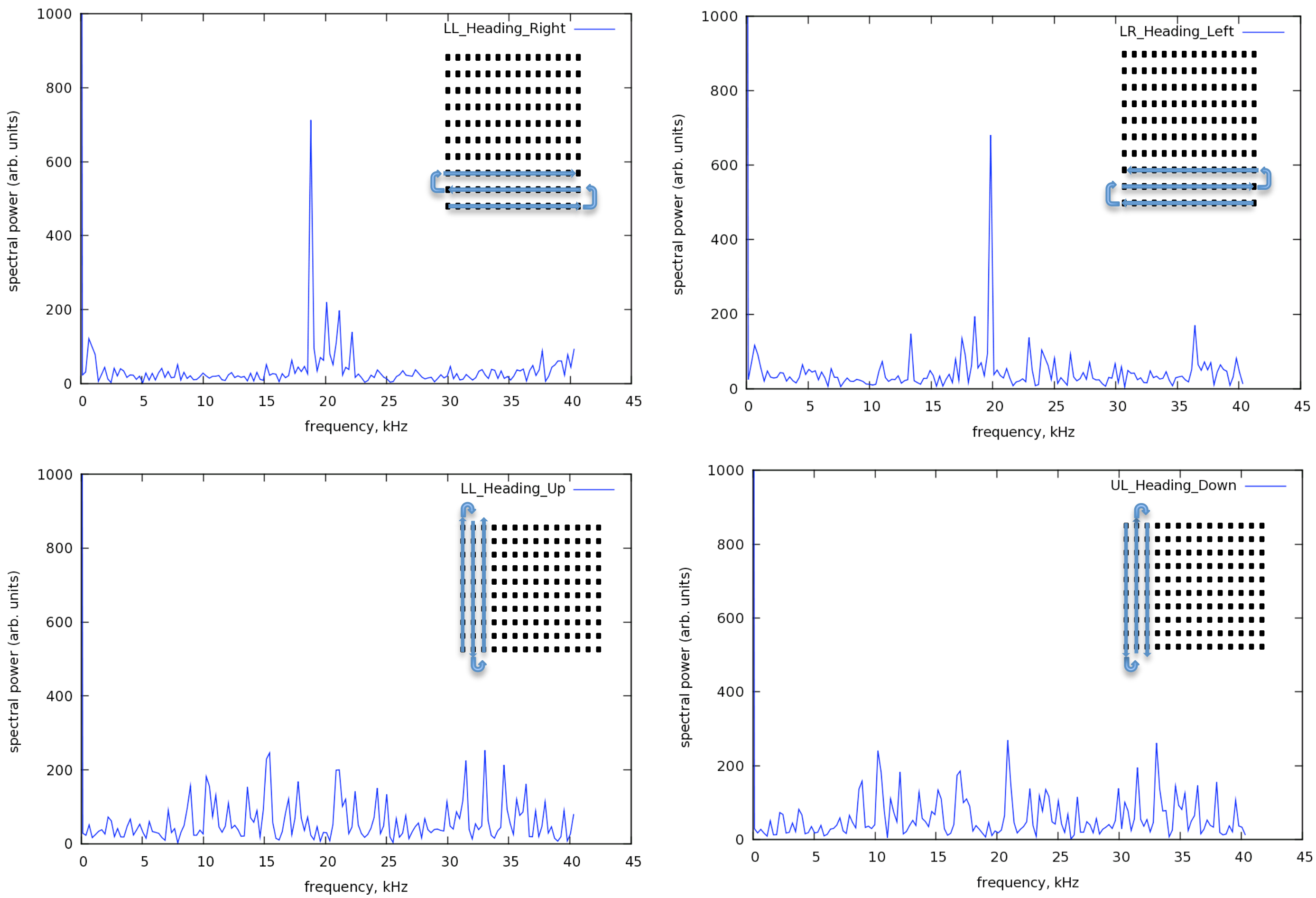
This patch of dots was written inside a single subfield, so we know that they were written consecutively. This is important if we want to calculate the noise spectrum.

Unfortunately, it is difficult to tell the order in which the dots were written.

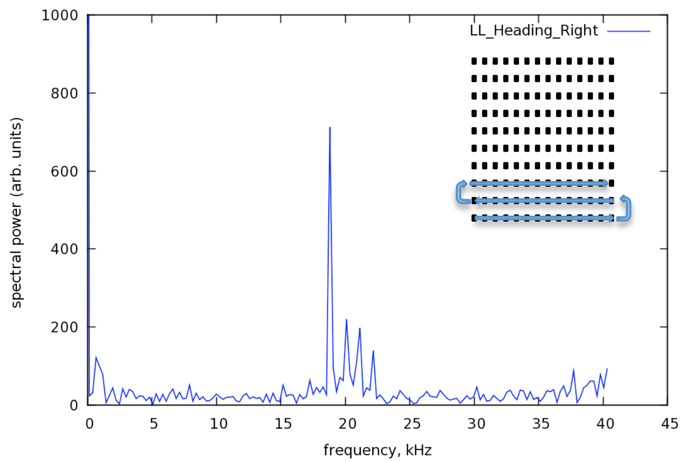
Why not calculate the noise spectrum?

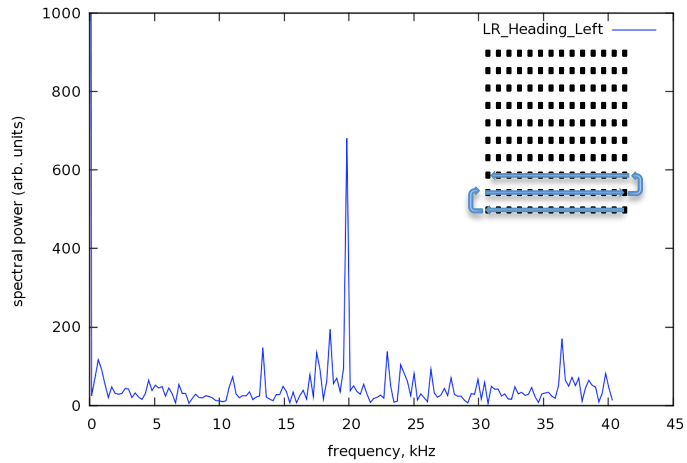
We must first arrange the dots according to the order in which they were written.

There are four possible writing orders, since time-reversed paths are equivalent.



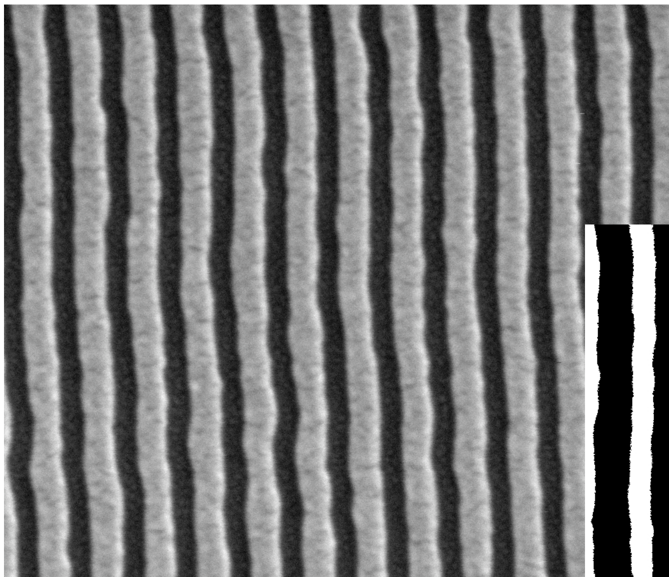
Why not calculate the noise spectrum?

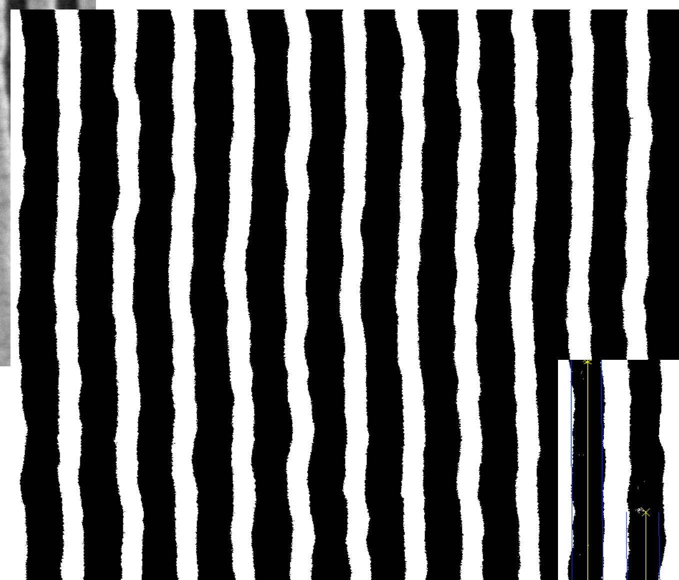




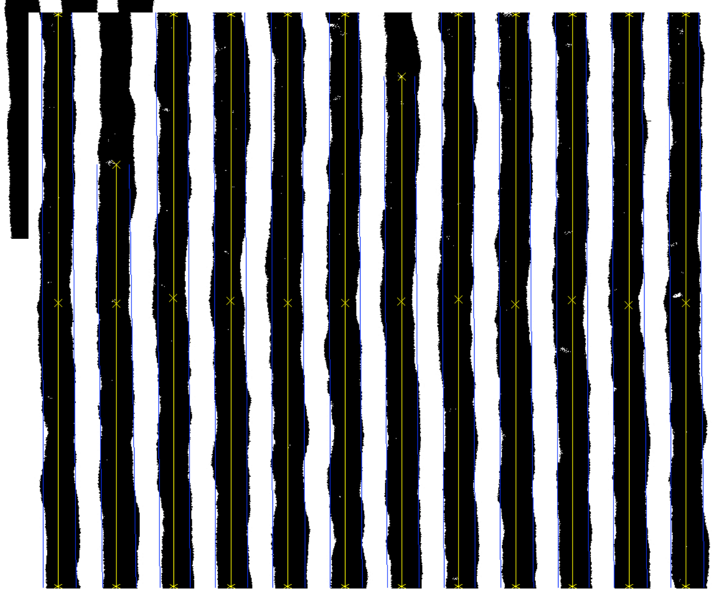
The vertical paths are clearly incorrect, but either horizontal path could be right. We would need an additional test frequency to sort it out. Spectra extracted from images such as these are highly dubious, since pattern generators will typically change the raster to match the longest side of the rectangle, and since the sample can be rotated arbitrarily in the SEM. Because of these problems, spectral analysis is not included in the current version.

Added bonus: line edge roughness program





3σ roughness = 12 nm



A very similar algorithm
can be used to find dark
lines on a bright background.
Straight lines are fit to
the edges, and the roughness
is calculated.

In addition to the usual
threshold function, the user
must be sure to cut the
line ends horizontally, so that
rounded ends do not contribute
to the roughness value.

What is this for? Who needs it?

When testing a new e-beam writer, this dot analysis technique provides a simple and unambiguous way of specifying and measuring the deflection noise. When diagnosing problems, it eliminates the convolutions of line-edge roughness. This program can be used to quantify the accuracy of a patterning instrument or the coherence of self-assembly.