This document contains instructions for using the MATLAB™ code used for quantifying transfer film free-space length for solid lubricants, as explained in the article[1] appearing in the scientific journal 'Wear'

J Ye, HS Khare, DL Burris

April, 2014

I. Using the code:

The following is a step-by-step instruction on using the dispersion code.

Step 1

[1] **Quantitative Characterization of Solid Lubricant Transfer Film Quality** J. Ye, H.S. Khare, D.L. Burris*, Wear XX (2014) XXX-XXX

Upon execution, the code will prompt for a bitmap of the transfer film image to be analyzed. This could be an optical, profilometric or electron microscopic image. Note that the format of this file should be **.bmp**. The image size (in pixels, or otherwise) or the aspect ratio is not important, however, **pixel values should be even** (for example, 231px x 422px will need to be cropped to 230px x 422px, etc.). The image *should* be in RGB scale (grey scale image need to be converted to RGB scale) and have a good contrast between film and substrate. If a converted black and white image is already available, answer 'Y' to the following question and skip to step 6.

```
Is the transfer film image converted to black and white already? (Y/N):
```

Step 2

In the next step, you are required to manually pick up reference points representing bare counterface in the transfer film image. For the best results, select the highest points on the surface (as opposed to scratches). The statistics for the color intensity of this sampling of points will be used to distinguish the counterface from the film. Enter the number of reference points you wish to use at the following prompt (we find that 20 points is a good compromise between time and a statistically robust result):

```
Enter number of counterface reference points for image conversion:
```

Step 3

In the next step, you are asked if the transfer film appears brighter or darker than the counterface in the provided image. This will help the code to determine the right thresholding direction. Answer 'Y' or 'N' at the following prompt:

```
Is the transfer film brighter than substrate? (Y/N):
```

Step 4

In the next step, you are asked to choose the number of standard deviations to use for defining color intensity bound for image thresholding (see Section II for more details). 3 to 5 is suggested as illustrated in the referenced paper.

```
Enter # of STDEVS to use for defining lower or upper bounds of color
threshold (Suggested = 3 to 5):
```

Step 5

In the next step, you are prompted with a converted black and white image based on previous selected reference points and number of standard deviations. Answer 'N' if the converted image appropriately represents the parent image. Answer 'Y' otherwise to return to step 4. If areas of counterface are being mistaken for transfer film, the number of standard deviations should be increased and vice versa.:

```
Tweak #STDEV further? (Y/N):
```

Step 6

Now you are required to provide the physical width of the transfer film image selected in step 1 to convert pixels into a unit of length. Enter the width of the transfer film image in micrometers at the following prompt:

```
Enter approximate width of transfer film image in micrometers:
```

Step 7

A starting estimate of the characteristic free-space length needs to be provided next. Visually estimate the average length-scale of the bare regions of counterface in micrometers and enter a ***larger*** value at the following prompt:

```
Enter initial guess of characteristic square width in micrometers:
```

In auto-mode, the code only iterates down until the mode is zero. Smaller input values will also have a mode of zero and the code will return your entered value. This indicates that a larger starting point is needed. Very large starting points will increase computational time. For suggestions on selecting the initial value, please refer to section II below.

Step 8

In order to generate histograms of pixel distribution within the chosen box size, you next need to specify how many of the boxes are to be used for gathering data for the histograms. Do this at the following prompt:

```
Enter number of random squares to use for analysis:
```

A higher number of boxes increases statistical robustness of your answer, but will take longer to converge. A smaller number of boxes increases speed but reduces accuracy.

We recommend using 100 boxes to quickly find a close starting point for a final analysis with 1000-5000 boxes. For suggestions on selecting the appropriate number of boxes, see section III

Step 9

After the automatic iteration to the characteristic box size is completed, the code prompts if you wish to proceed to the 'Manual Mode'. The manual mode is an optional extension of the code that allows you to fine-tune the result. If you wish you use the value generated by the automatic iteration, and not enter the manual mode, enter 'n' at the following prompt:

```
Automatic iteration completed. Continue to manual mode? (Y/N)
```

More information on the Manual Mode is given in Section IV

Results:

The final value of transfer film free space length (L_f) after iteration is displayed:

```
Value of free space length is (in micrometers):
```

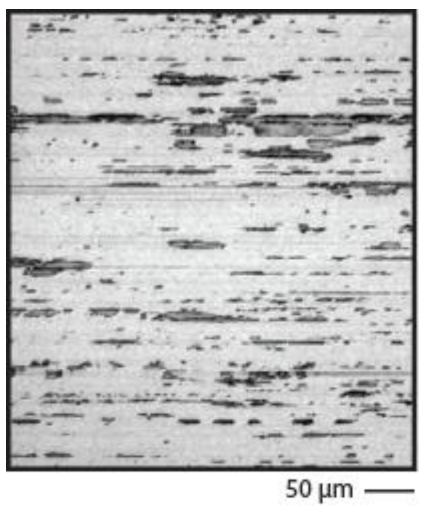
Further, the area fraction of the image occupied by transfer film (identified as black pixels) is also displayed:

```
Film area fraction (where film is represented by black pixels):
```

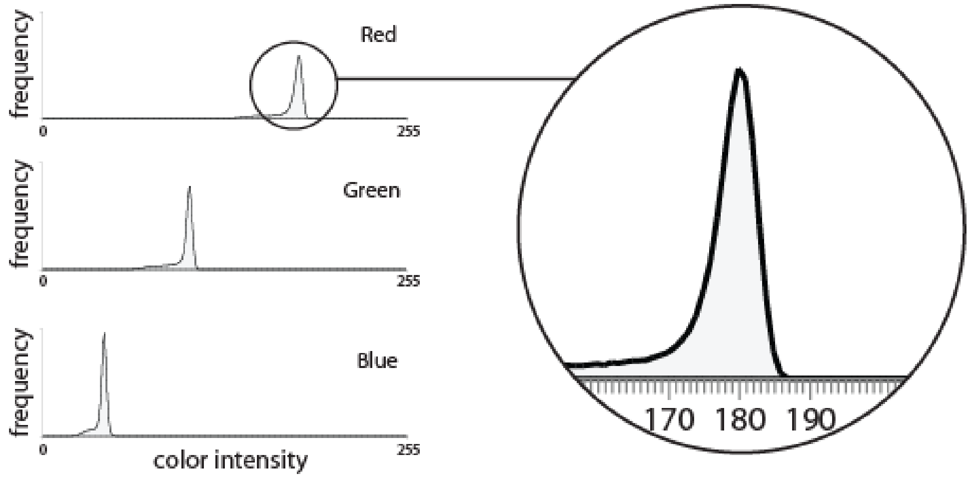
A histogram is generated showing the zero-mode corresponding to the free space length displayed above.

II. Choosing representative transfer film images for analysis (step 1)

Before applying the code, you need to choose appropriate transfer film images used for analysis. The code can be used on optical micrographs, electron micrographs and profilometric maps depending on what you have access to. Image should be **representative** of a certain transfer film morphology and also provide **good contrast** between transfer film and counterface. We have found that the image width should be on the order of 10x the free-space length. When the image is much larger, it either lacks resolution or takes excessive time to analyze. When it is much smaller, the analysis lacks statistical robustness. Below is an example image of a transfer film with an appropriately scaled field of view.

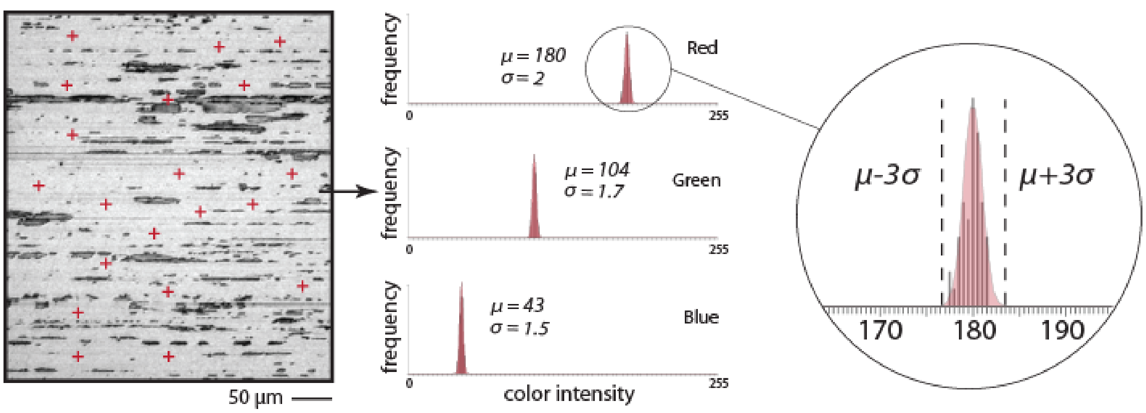


This image is an optical micrograph of a low wear alumina-PTFE nanocomposite transfer film taken with a bright field, reflected light microscope (Nikon MM-400/S). The counterface was pre-polished to 10 nm Ra and therefore reflects light much stronger than transfer film and has a brighter appearance than film. Dimension of the image is 440 (W) × 498(H) pixel. Below is the color intensity histogram of all pixels within the image. Notice the long tail at the left of each plot which represents the transfer film covered regions.



III. Selecting counterface reference points for image conversion (step 2)

In step 2, you are prompted to manually pick up a statistically significant number of random pixels where transfer film is known to be absent. The rationale is to distinguish the film free counterface area by calculating the channel mean and standard deviation of all pixels within this region.



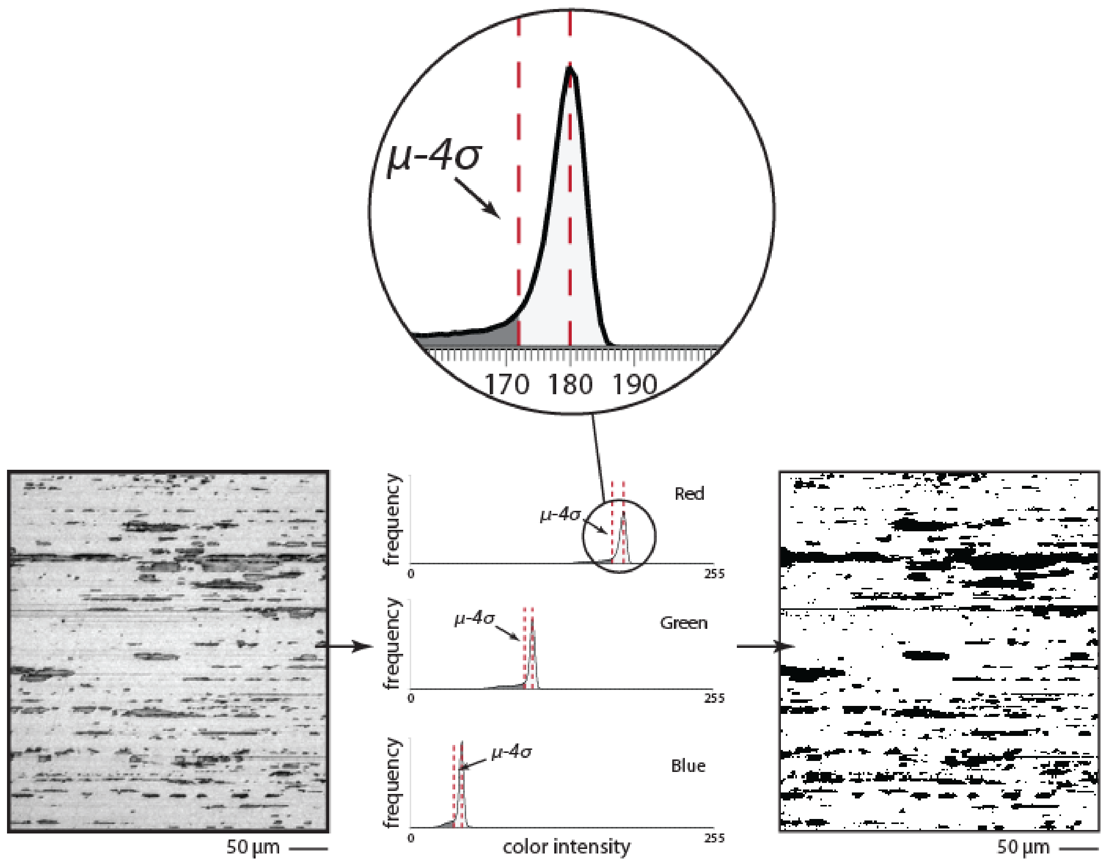
In the example image shown above, we manually select 50 points as references. Histograms of the color intensities of selected points were shown. It can be seen that each distribution can be satisfactorily described using Normal Distribution:

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where the mean (μ) and standard deviation (σ) of the distribution can be estimated from sampled dataset. The simulated full distribution curve (shadowed) were overlaid on top of the sampled data as shown above. In addition, such simulated histograms of color intensity closely resembled the high peaks in the histograms of the overall image (see figure in previous page). This provides the necessary condition for thresholding the image based on pixel intensities.

IV. Selecting number of standard deviations (step 4)

In step 4, the user is prompted to choose number of standard deviations (nσ) used for defining the RGB color threshold during image conversion ($\mu \pm n\sigma$)[2]. As seen in previous section, pixels of counterface are known to be within $\mu \pm 3\sigma$ range with a 99.7% confidence. By declaring any pixel with intensity outside this range to be covered with transfer film will provide a reasonable conversion of the original RGB image into a binary matrix. In our experience, we find that 3-5σ consistently provides optimal detection. In the example image shown below, n = 4 results in a binary matrix that closely resembles the appearance of its parent image. Here, **black** represents film and **white** represents uncovered counterface.

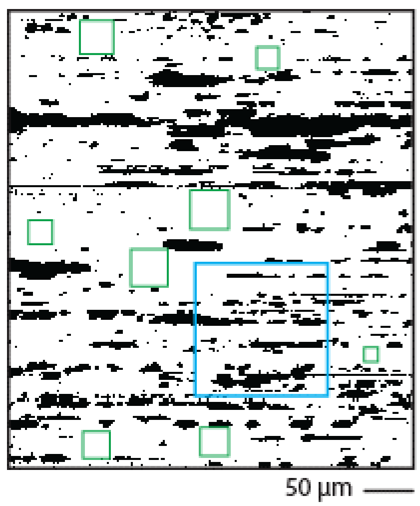


[2] Depending on the image histogram, 'μ - nσ' (when transfer film is darker than counterface) or 'μ + nσ' (when transfer film is brighter than counterface) will be set as the threshold value.

V. Selecting size of initial box (step 7)

At step 7, you are prompted to enter an 'initial guess of characteristic square width in micrometers'.

When using this code to calculate the size of the free space length, we are interested in finding the (largest) square size for which the most likely number of intersecting black pixels in a randomly placed square is zero. The code uses an initial user-input, and iterates to the actual value by generating these boxes, and placing them over across the transfer film image. A good analogy is to compare it to throwing a fishing net at different points in a lake, with the hope of *not* catching any fish. Of course, you will need to keep reducing the size of your net, until your reach a 'critical' dimension where you're more probable to drop the net at different locations, and still not catch anything. The largest net for which you are most likely to catch no fish is the free space length. Looking at the transfer film image then, we need a subjective measure of what *seems* like the free space length. The green boxes in the following image could all very well be this starting size:



Note that a box size similar to the blue box would never be completely particle-free, regardless of where you position it. We seek to identify which among the green boxes is the truest representation of the free space length.

For this purpose, you should select a numeric value slightly ***larger*** than the apparent free-space length (green boxes in this case). The code iterates to smaller sizes until it finds the free-space length. Alternatively, you may use a value which *seems* like the free space length (or even slightly smaller) and if you're too small, you can always start again with a marginally increased value (a good way to tell if you're too small is if the value the code outputs is the same as the value you entered for the box size). The disadvantage in using the former is wastage of computational time/effort. Since the code will be trying to work its way down from your starting value, depending how overestimated this starting value is, the code might take a prohibitively longer time simply an as artifact of the starting choice. For this purpose then, a size slightly smaller than the perceived free space length is preferred because if it's too small, the code will "tell" us, and we can always restart with a slightly incremented starting value.

For the illustration shown above, then, the width of the transfer film image were 406 micrometers, the size of our initial guess (which should be definitely less than the size of the blue box), would be about a fourth of the image width (or maybe smaller still), or roughly 100 micrometers. If your initial size is too small (say, the same size as one of the smaller green boxes), you can restart with a larger initial guess.

VI. Selecting the number of boxes (step 8)

In order for a given sized square to be indicative of the free space length, it is imperative that the given square is representative of the film spatial statistics across the entire image. As a result, it is not difficult to realize that for a more accurate and statistically robust measurement of the characteristic box, a larger number of squares need to be overlaid at random locations on the image. While an increase in number of boxes overlaid (for any box size) increases accuracy (how close the number you're getting is to the actual free space length) of the eventual 'solution', increasing the number of boxes may increase computational effort.

It is observed that, usually, the same computational effort is required for a larger number of smaller sized squares (say, an order of magnitude smaller than the image size), as is for a smaller number of large-sized squares (say, comparable or of the same order of magnitude as the image size). The relative importance of computational speed and accuracy of result is the choice of the user.

For images where a larger starting box size is necessitated by nature of transfer film distribution, it is advisable to limit the number of squares in the initial stages. Thereafter, depending on the convergence speed, the user may subsequently increase the number of squares to obtain a more accurate solution.

Always bear in mind, however, that smaller sampling will always give you numbers which will be off substantially from the actual characteristic box. If the code is run multiple times on the same image (say, in manual mode), with the same parameters, changing only the number of boxes, a large scatter in your free space value will be seen for a smaller 'number of boxes' value. This is largely mitigated at higher 'number of boxes' values.

Ideally, a user may initially wish to experiment with box numbers between 100 and 1000, to obtain the required accuracy coupled with desired convergence speed. As mentioned earlier, if the square size is large relative to the image size, one may start with box numbers as few as 50, and work their way upwards till computational time becomes prohibitive. Once an accurate-enough value for characteristic box is obtained (reduced scatter between successive runs), 'fine tuning' to get the exact value may be done in the manual mode.

VII. Manual Mode (step 9)

After the code iterates to the value of the characteristic box size, an option is provided by which the user can enter a 'manual' mode. Using the value provided by the automatic mode, the user can, without having to reselect the image, input only the new characteristic size and number of boxes he or she wishes to simulate, and obtain results for those parameters.

The manual mode only provides the film pixel distribution for the given box size and box numbers- it does not attempt to iterate on its own to an 'optimal' value. This feature is helpful for those users who wish to 'fine tune' the value provided by the automatic iteration, or those who wish to gather information on how changing an individual parameter changes the output (for instance, how changing the number of boxes for the same box size changes the shape of the histogram).

'Fine tuning' may be performed by using as a reference the final value outputted by the automatic iteration (provided care was taken to use an adequate sampling size to reduce scatter). Using this value, the user may manually increment or decrement in the box size and observe the generated histogram to ascertain the value of the characteristic box size. Further, since no step-wise iteration is performed, a single 'manual-mode' measurement is significantly faster than the automatic mode, which means that a higher sampling size (number of boxes) may be used to truly iron out the real characteristic box size.

As an illustration, one may use 200 boxes for a given rough starting box size, work up to 500 boxes till computation time becomes prohibitive. Then, using the value the code gives you at 500 boxes, and incrementing/decrementing that value with a much higher number of boxes (say, even 3000, etc.) one can find the size of the characteristic box to a resolution of a single nanometer (this, of course, would be a function of statistical error derived from your sampling size).

It may be mentioned here that the only data that 'carries over' from the automatic mode to the manual mode is the image pixel matrix, and the image size in nanometers. Aside from these, values you obtained from the automatic mode (box size, histogram, etc.) do not influence your results in the manual mode. The auto mode can be thought of as merely something to give a reference point for starting the manual tuning.

VIII. Illustrations

In this section, a walk-through is provided for measuring the free space length of two transfer film images: the first is an optical image of a low wear alumina-PTFE nanocomposite transfer film taken with a bright field, reflected light microscope (Nikon MM-400/S). The second is a profilometry image of transfer film of the same material system taken with a scanning white light interferometer (Wyko NT9100).

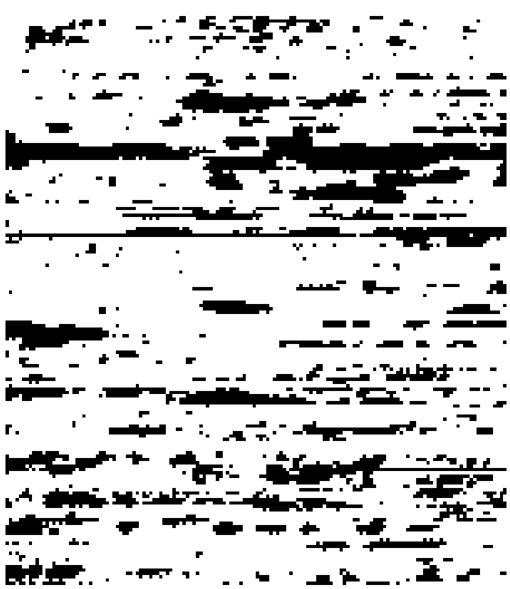
Optical Image of Transfer Film

In the illustrative, the example optical image was shown on the right, we notice that transfer film is composed of many streak-like debris spreading across the view field and has slightly darker appearance. They are distributed rather uniformly despite most of the area is still uncovered and is transfer film free.

Our aim in this example is to measure the free space length of such transfer film distribution. It is assumed that such metric is important to the wear reduction induced by any transfer film at sliding surfaces.

First, let us establish some parameters. The width of the image, as we can calculate from the scale bar given, is about 406 µm. In an actual transfer film micrograph (as we shall see in the next example), the size of the image is often known. The steps involved in evaluating the free space length (L_f) are as listed below:

1. Executing the .m file, at the prompt: `Is the version of MATLAB 7.0 or later? (Y/N) :` we enter: `Y`
2. At the following prompt: `Is the transfer film image converted to black and white already? (Y/N):` as we don't have a converted image already, we enter: `N` and select the sample image 'Optical.bmp'
3. The following prompt asks us to `Enter number of counterface reference points for image conversion:` we enter `20` in order to save some labor and still be able to pick up the statistics of pixel intensity distribution. A window will pop out displaying the raw image, waiting for mouse input. We go ahead and select 20 points at transfer film free area across the whole view field
4. The code now asks a question: `Is the transfer film brighter than substrate? (Y/N):` we enter `N` as the transfer film appears darker than the substrate in our case
5. At the following prompt, we are asked to `Enter # of STDEVS to use for defining lower or upper bounds of color threshold (Suggested = 3 to 5):` we enter `4` as we discussed earlier in section III. It takes a few seconds before MATLAB finish the image conversion and pop out the converted binary image (shown below)
6. At the following prompt: `Tweak #STDEV further? (Y/N):` we enter `N` as this converted image is closely representative of the original image, if the you are not satisfied with the conversion, enter `Y` and you will be able to choose a different number of standard deviation used for thresholding in previous step



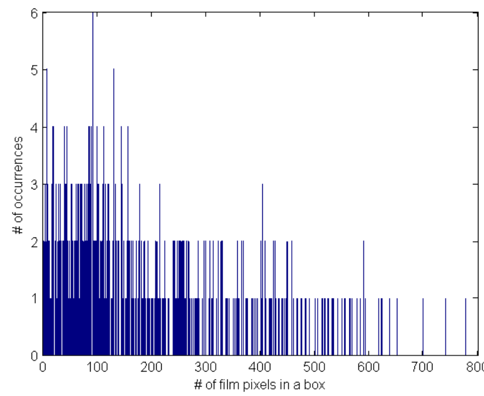
7. The following prompt asks us to `Enter approximate width of transfer film image in micrometers:` we enter `406` as we calculated from the scale bar earlier
8. The following prompt asks us to `Enter initial guess of characteristic square width in micrometers:` Observing that the image width is 406 µm, and as per the discussion is section V, the pockets of filler-free space

seem to be about a fourth of the image width, we enter `100` µm as our initial guess (underestimate the one-fourth value).

9. Next, we are asked to `Enter number of random squares to use for analysis:` Since we are unsure of how computationally intensive our selection of 100 µm is going to be, we choose to use 500 squares.

10. After allowing the computation to finish (in this case, the simulation ran for less than 5 seconds, while this may vary from one run to another even with same parameters), we obtain the following results:

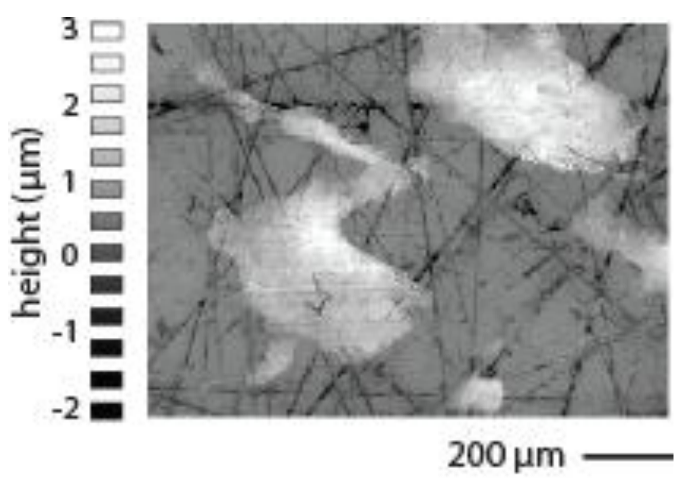
```
Value of free space length is
(in micrometers):
27
Film area fraction (where film
is represented by black pixels):
0.2163
```



Further, the adjoining histogram is generated. Clearly then (and as was discussed in section II), our starting guess of 100 µm was a large overestimate of the actual 27 µm. While not illustrated in this example, the value of 27 µm may be further tuned in manual mode.

Profilometry Image of Transfer Film

In this example, we will use a profilometry image of transfer film shown in the right. Notice that transfer film is composed of two large, discrete pieces of debris and a few other smaller debris in the view field.



Our aim in this example is to demonstrate measuring the free space length using a profilometry image of transfer film. More importantly, the use of profilometry image provide an unique advantage of distinguishing film from rough counterface as counterface scratches can become indistinguishable from the film under optical mircoscopy. While in a profilometry map, scratches will always be less intensive than the film and can be thresholded easily.

First, the image width, as can be calculated from the scale bar, is about 1.15 mm, or 1150 µm. This can also be obtained from the interferometry software used for taking the image. The steps involved in evaluating the free space length (L_f) are as listed below:

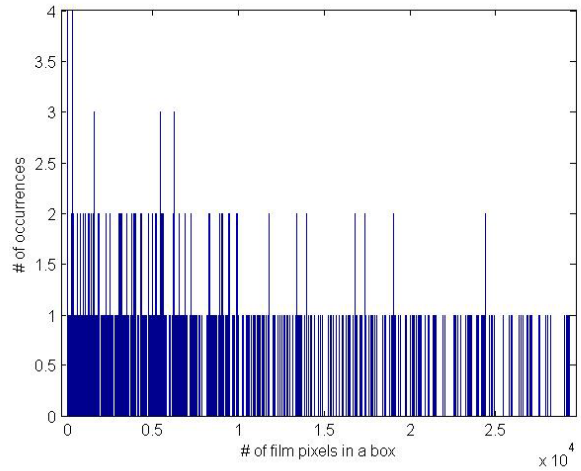
1. Executing the .m file, at the prompt: `Is the version of MATLAB 7.0 or later? (Y/N) :` we enter: `Y`
2. At the following prompt: `Is the transfer film image converted to black and white already? (Y/N):` as we don't have a converted image already, we enter: `N` and select the sample image 'HeightMap.bmp'
3. The following prompt asks us to `Enter number of counterface reference points for image conversion:` we enter `20` in order to save some labor and still be able to pick up the statistics of pixel intensity distribution. A window will pop out displaying the raw image, waiting for mouse input. We go ahead and select 20 points at transfer film free area across the whole view field, pay attention not to select points at those dark lines which are deep scratches as they only comprise a small portion of the counterface and are not representative to the whole transfer film free domain
4. The code now asks a question: `Is the transfer film brighter than substrate? (Y/N):` we enter `Y` as the transfer film does appear brighter than the substrate in this case
5. At the following prompt, we are asked to `Enter # of STDEVS to use for defining lower or upper bounds of color threshold (Suggested = 3 to 5):` we enter `4` as we discussed earlier in section III. It takes less than half minute before MATLAB finish the image conversion and pop out the converted binary image (shown below)

6. At the following prompt: `Tweak #STDEV further? (Y/N):` we enter `N` as this converted image is closely representative of the original image, if the you are not satisfied with the conversion, enter `Y` and you will be able to choose a different number of standard deviation used for thresholding in previous step
7. The following prompt asks us to `Enter approximate width of transfer film image in micrometers:` we enter `1150` as we estimated earlier
8. The following prompt asks us to `Enter initial guess of characteristic square width in micrometers:` Observing that the image width is 1150 µm, and as per the discussion is section V, the pockets of filler-free space seem to be about half of the image width, we enter `550` µm as our initial guess (underestimate the one half value).
9. Next, we are asked to `Enter number of random squares to use for analysis:` Since we are unsure of how computationally intensive our selection of

100 µm is going to be, we choose to use 500 squares (which is about halfway between the recommended 100 and 1000)

10. After allowing the computation to finish (in this case, the simulation ran for less than a minute, while this may vary from one run to another even with same parameters), we obtain the following results:

```
Value of free space length
is (in micrometers):
263
Film area fraction (where
film is represented by
black pixels):
0.2559
```

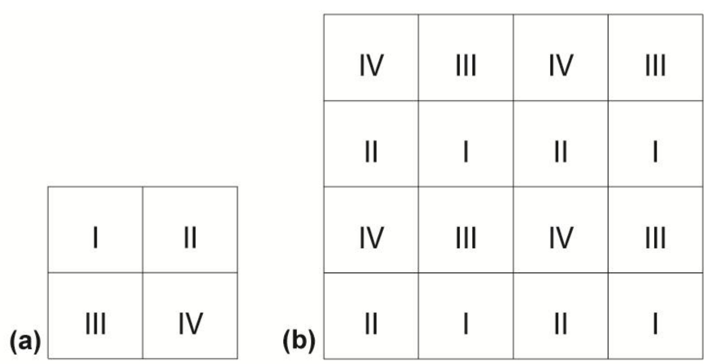


Further, the adjoining histogram is generated. Clearly then (and as was discussed in section II), our starting guess of 550 µm was a little overestimate of the actual 263 µm. While not illustrated in this example, the value of 263 µm may be further tuned in manual mode

IX. Overview of code algorithm:

The general algorithm of the code is as outlined:

1. A bitmap (.bmp) transfer film image (RGB mode, any pixel size, any aspect ratio) is imported to MATLAB^{TM}, and stored as matrix ‘AA’, it’s elements being the R,G,B values associated with the pixels in the image.
2. After user manually picked the reference points at counterface area, mean (μ) and standard deviation (σ) of pixel values at selected positions were calculated for R, G and B channels.
3. With a threshold pixel value of ($\mu \pm n\sigma$) to distinguish film from substrate, pixel with an intensity value outside this range will be declared as transfer film free and stored in a new matrix ‘A’ with pixel value ’0’, while pixel at transfer film covered region will be given a pixel value ‘255’, the resulting matrix A will be solely composed of 0 and 255 and can be displayed as a binary image
4. With a threshold grayscale value of 150 to distinguish film from the substrate, transfer film area fraction (‘af’) is evaluated.
5. A ‘data’ matrix is next generated, which has periodic boundary conditions. In other words, the data matrix is composed of the ‘A’ pixel matrix at the core, adjoined by portions of the ‘A’ matrix such that a periodic boundary is obtained at the edge of the ‘data’ matrix. This is illustrated in figure A1, where I, II, III, IV indicate the quadrants of the original ‘A’ matrix.
6. An initial value of box size is inputted from the user, together with the number of random boxes that the user wishes to analyze the image with and the size of the TEM image in nanometers.
7. The code converges on the characteristic box size, thereafter performing incremental ‘jump ups’ and ‘trim downs’ to converge to the actual characteristic box size. If the distribution is bimodal, the ‘false’ mode is suppressed in order to obtain the correct value of ‘l’.
8. Results are displayed, together with a histogram of particle occurrences for the given ‘Nsquare’ boxes.
9. A Manual mode is provided at the end of the code, which in essence has the similar algorithm as the computation above, except for the automatic box-size adjustments (jump up/trim down) which cause the code to iterate to the characteristic value. Instead, the manual mode just provides the histogram for the value of box size and sampling size selected by the user.



(a) The image pixel-matrix ‘A’, with its four quadrants indicated by numerals. (b) The ‘data’ matrix, with ‘A’ at its core, and periodic boundaries along the edges. Note that if ‘A’ is of order 500x500 (say), the ‘data’ matrix is of order 1000x1000.