A Free Matlab Script for Spatial Drift Correction

Joshua D. Sugar,[1]* Aron W. Cummings,[1] Benjamin W. Jacobs,[2] and David B. Robinson[1]

[1]Sandia National Laboratories, P.O. Box 969, Livermore, CA 94551
[2]Protochips, Inc., 616 Hutton St., Suite 103, Raleigh, NC 27606

*jdsugar@sandia.gov

Introduction

One of the simplest operations to perform on frames of video data is a translation to align features in adjacent frames. The vast quantities of data generated in video output from an *in situ* transmission electron microscopy (TEM) experiment, for example, require that data processing be performed without operator intervention. Then the operator has more time to enjoy the finer things in life. We have automated the process of feature alignment in the script we describe here.

In order to quantitatively measure or track changing features during a reaction, it is necessary that the features of interest do not move from frame to frame of video data. However, this can be difficult to control when performing an *in situ* experiment, especially a heating experiment. An example of the measured drift that occurs while heating a sample *in situ* is shown in Figure 1. At temperatures ranging between room temperature and 1000°C, the sample can drift by as many as tens of micrometers depending on the heating technology used. Even a drift of a few µm, however, in the case of a microfabricated heater [1–3], is enough to move the area of interest out of the field of view at intermediate (20–80 kX) TEM magnifications. Drift can occur in these experiments for a variety of reasons. These include thermal expansion of the sample and/or sample holder during heating, pressure-induced deformation of environmental cells, or motion of the sample itself as a result of environmental stimulus or interaction with the electron beam. Although strategies are available to minimize drift, it is difficult to reduce to zero.

Ideally, one would predict the sample motion and adjust the sample position during an experiment so that each frame is aligned with its previous one. The approach presented here could be applied to such a real-time feedback scheme, but this would require computer actuation of the sample position, which involves communication between our drift-correction software and the microscope stage software. Because of the several manufacturers of microscope platforms and software controls, it is not possible to easily design a drift-correction package that works for all hardware and software combinations. Therefore, a post-processing step that aligns the video frames after completion of the experiment is desirable. Our goal was to develop a simple script that is freely available and can automatically correct spatial drift in video files for a variety of formats. While the Matlab platform is not free, it is common, and the script is simple enough that it could be ported to other platforms that can encode and decode image and video files. This article describes the Matlab script and how to use it.

Description of Cross-Correlation Algorithm in Matlab

There are several spatial drift algorithms that exist in other aspects of electron microscopy. The commonly used
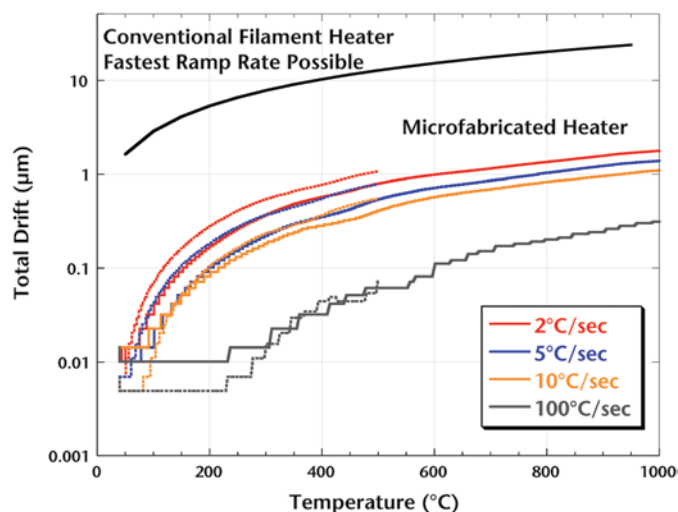

Figure 1: Sample drift data for various heating rates to a maximum temperature of 500°C and 1,000°C. A sample heated by a local filament can drift as many as tens of microns during an experiment. A microfabricated heater is less prone to large drifts, but still moves by several microns, which can be significant depending on the magnification. It drifts less at higher heating rate because there is less time to heat areas surrounding the sample during the experiment.

software packages for data acquisition and analysis, such as Digital Micrograph (Gatan, Inc.), TIA (FEI), INCA (Oxford, Inc.), and Espirit (Bruker GmBH) (just to name a few), implement spatial drift correction routines when acquiring a spectrum image or energy-filtered elemental maps. In fact, a sophisticated and excellent routine for correcting energy-filtered spectrum images (EFTEM-SI) is freely available [4] and can correct files in the Digital Micrograph ".dm3" format. However, that routine is not easily reconfigured to correct drift in files of any arbitrary format, and its original intent was for use specifically with EFTEM-SI data. Here we build on existing techniques and extend the methods used to correct drift in other applications to video files of an unspecified format.

The cross-correlation method for calculating the shift needed to align images is relatively straightforward [5]. The cross-correlation c_i for frame f_i and reference frame f_{ref} is

$$c_i(x,y) = \iint f_{ref}(x',y') f_i(x+x', y+x') dx'dy'. \qquad (1)$$

For frames composed of discrete pixels, the integral can be replaced with a sum. The frames are taken to be periodic, so that $f(x,y) = f(x+nW, y+mH)$ for any integers n and m, where W is the image width and H is the image height. For the simplest case, we assume that the frame in question, f_i, is essentially identical to the reference frame, f_{ref}, except for a shift by some amount (a,b), i.e. $f_i = f_{ref}(x–a, y–b)$. Then the value of c_i is

maximized at a certain pair of values *a* and *b* that best correct for the drift, so the shifted f_i best matches f_{ref}. The cross-correlation can be efficiently determined by considering the two-dimensional Fourier transform of (1), as given by the convolution theorem:

$$C_i\left(\omega_x,\omega_y\right)=F_{ref}^*\left(\omega_x,\omega_y\right)F_i\left(\omega_x,\omega_y\right). \qquad (2)$$

We can then compute the inverse Fourier transform of (2) to obtain c_i, and *a* and *b* are determined from the position of the maximum of c_i. If a given frame is exactly like the reference frame except for the shift, the maximum will be very sharp. If there are some differences between the images, the maximum will be less sharp, but still identifiable in many practical situations.

This approach can be easily implemented in a variety of commercial software packages, and we have chosen Matlab. The first step, as indicated above, is to find the product of the FFT of the shifted image ($Image_{drift}$) and the reference image ($Image_{ref}$). To find the cross-correlation function for these two images, the Matlab code would look like the following:

$$\text{fft\_prod} = \text{fft2}\left(Image_{drift}\right).*\text{conj}\left(\text{fft2}\left(Image_{ref}\right)\right);$$
$$\text{cc} = \text{ifft2}(\text{fft\_prod}); \qquad (3)$$

The variable cc is a two-dimensional matrix that contains the values of the cross-correlation function. In order to find the appropriate shift amount for $Image_{shift}$ so that it aligns with $Image_{ref}$, we just have to find the maximum of the cc variable. This can be done with the command

$$[\text{yshift}, \text{xshift}] = \text{find}\left((\text{cc}) == \max(\max(\text{cc}))\right) \qquad (4)$$

The final step is to shift $Image_{drift}$ so that its features are aligned with $Image_{ref}$. There are several ways to perform this step, and we will describe two in the next section. One way to shift the images without having to increase the number of pixels is to use the command *circshift*. Then, the new aligned image $Image_{aligned}$ becomes

$$Image_{aligned} = \text{circshift}\left(Image_{drift},[\text{height-yshift},\text{width-xshift}]\right). \qquad (5)$$

This procedure is illustrated in Figure 2. A feature of interest is located and imaged in (a). During an experiment, the feature drifts to the position shown in the image (b). To correct for the image drift, the cross-correlation of (a) and (b) is calculated and plotted in (d). The position of the maximum of (d) relative to the image origin indicates the number of pixels the image in (b) must be shifted to be aligned with the original image (a). After applying the appropriate shift, the image in (c) is produced, and now the features in (a) and (c) are spatially aligned.

Free Download of MATLAB Script

We have written a Matlab script that performs these actions on an arbitrary data set in a variety of formats. Our script, *drifty_shifty_deluxe.m* is freely available on the Matlab File Exchange website at http://www.mathworks.com/matlabcentral/fileexchange/45453-drifty-shifty-deluxe-m. In the next section, we will provide a brief explanation of how to use this script.

Instructions for drifty_shifty_deluxe.m

In order to use *drifty_shifty_deluxe.m* to correct the spatial drift in a series of images, the data must be in a format that can be read. It is possible to use a series of .png images, a series of .tif images, or a movie file. The ability of Matlab to read movie files depends very much on the platform being used (Windows or Mac) and the codecs that are available on your computer and installed properly for Matlab (see reference [6] for details). We have found that .mp4 and .mov files work with Matlab on a Mac. However, these files don't always work on a PC. Therefore, some attention is needed when deciding the format of video used for data collection. It may be necessary to convert from one video format to another, which can be time-consuming. In these cases, it may be necessary to use applications like Quicktime Pro (Apple), Flip4mac Studio (Telestream), BTV pro (Bensoftware), Windows movie maker (Microsoft), or VLMC (when available, VideoLAN Organization) to convert video formats or combine different movie files into one larger file. In general, .avi files seem to be consistent across various platforms, but it
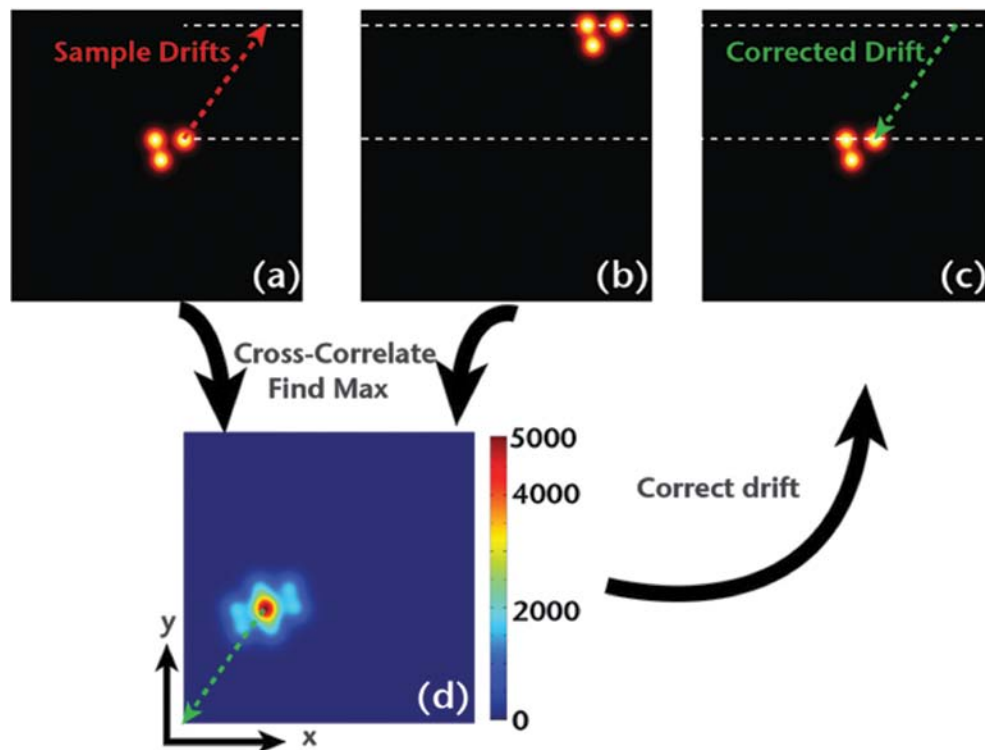

Figure 2: A schematic illustration of the procedures for correcting spatial drift in a series of images. The starting features are shown in (a). During an experiment, those features drift to the position shown in (b). The cross-correlation function is calculated for the images in (a) and (b) and is plotted in (d). The position of the maximum in (d) provides the information for the shift required to align the images because the maximum should be at the image origin. The appropriate shift is applied in (b), and the resulting corrected image is shown in (c).

is necessary to use Quicktime 7 on a Mac to display them properly.

With a video file or a series of images that can be displayed properly on a computer and read into Matlab, *drifty_shifty_deluxe.m* can be used to correct the drift. If the data contains annotations (scale bar, magnification, etc.), it will be better to remove them for drift correction because they do not move from frame to frame, and they can confuse the cross-correlation algorithm. In addition, the script works best with data that drifts smoothly during an experiment. If the data jumps randomly in the field of view, there could be some problems with aspects of the correction. When running *drifty_shifty_deluxe.m* there are a number of dialog boxes that pop up and let the user decide how to treat the data. Examples of the dialogs are shown in Figure 3, and each one is described here.

1. First, open Matlab and run *drifty_shifty_deluxe.m,* either by typing the script name in the command line or by running it from the script editor (the run button or ctrl+return on PC/command+return on Mac). Note that in order to run properly, the script must be in the current working directory of Matlab.

2. The first dialog box (Figure 3a) asks for the format of the input data. This can be a movie file, a series of .png images, or a series of .tif images. For .png or .tif files, the frames must be in numerical order when they are sorted by name.

3. Step 3 is to locate the data. For .png or .tif files, you can simply select the folder that contains all of the images. For a movie file, select the appropriate movie file.

4. The next dialog (Figure 3b) will ask "Pad frame with empty space, or wrap within same bounds?" The "pad" option will add black pixels around the image so that it may move within a larger frame in order to keep the feature of interest in the same region. The "wrap" option will put periodic boundaries on the images so that the feature does not move from frame to frame after drift correction and so the frame size is the same as the original image size. Examples of these two options are shown in Figure 5.

5. If desired, it is possible to average some input frames so they are combined into one frame of the output file. This dialog is shown in Figure 3c. The available options are 1 (no frame averaging), 3, or 5 frames.

6. The next option is to choose how large the drift-corrected output file should be (Figure 3d). The output files are uncompressed .avi files, and they can be quite large. Older versions of Matlab have limits on how large certain variables can be, and 2,000 frames per output movie seems to be agreeable when size limitations occur. The larger the size of the output file, the longer the calculation will take. Several minutes to tens of minutes is typical for movies with 20,000 frames. If the input file is 10,000 frames and the output file size is chosen to be 2,000, then the output will consist of five avi files, each of 2,000 frames (with an appropriate index in the file name). It will be necessary to stitch these files together in another program, such as Windows Movie Maker or Quicktime Pro, in order to have the full drift-corrected movie in one file.

7. The final step is to locate the Matlab data file that contains the drift data. After completing step 3, Matlab will provide you the location of the data file in the command window. It should have the form of *inputfilename_driftdata.mat* where *inputfilename* refers to the name of the file selected in step 2. If desired, it is possible to look at the drift data later, make changes to it, and recalculate the corrected movies. The corrected data will be output as *inputfilenamecorrected_index.avi*, where the index value indicates the order of the output files so that they are synced with the input movie file.

At this point, the drift-corrected output file can be used to calculate a morphologic property of a feature in the image because it will stay in the same location from frame to frame in the corrected data set. Note that on a Mac, the output .avi file needs to be played in Quicktime 7 to display properly. It is also possible to use the drift data directly if that contains useful quantities for the experiment (see below).

Additional Comments

There are a few additional features of *drifty_shifty_deluxe.m* that are worth mentioning. First, the drift data is stored in a Matlab data file. This means that it is possible to look at the drift data after the calculation is complete and plot it if desired. The drift data will contain shiftx and shifty values (in number of pixels) for each frame of the input movie file. With an appropriate scale
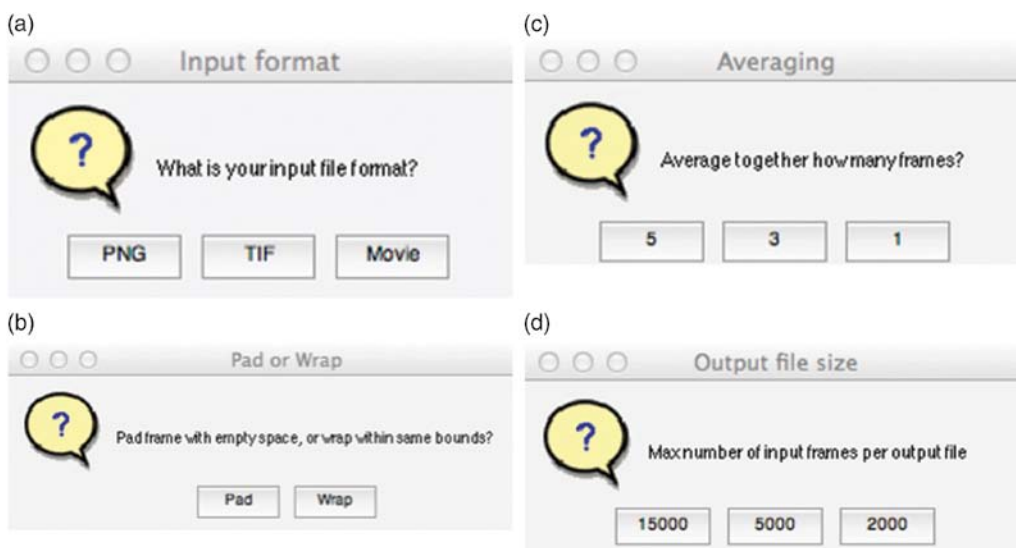

Figure 3: The series of pop-up dialog boxes used when running *drifty_shifty_deluxe.m*. The input file type is chosen in (a), the "pad" or "wrap" option in (b), the number of frames to average in (c), and the size of the corrected output files in (d).

```
03222012\Drift Data No FFT SHIFT                          ▼  ...  🔼

rgets.map'], 'general_rn_highlights');

ⓘ New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> drifty_shifty_deluxe
  General Settings:
     Duration = 420.7680
     Name = 008-RTto1000toRT 2xbin 20kx.wmv
     Path = E:\Drift Expts 03222012\Drift Data No FFT SHIFT
     Tag =
     Type = VideoReader
     UserData = []

  Video Settings:
     BitsPerPixel = 24
     FrameRate = 30
     Height = 480                          I
     NumberOfFrames = 8507
     VideoFormat = RGB24
     Width = 640


500 frames of 8507 done
8500 frames of 8507 done
I finished calculating the drift data.  It was saved to:
E:\Drift Expts 03222012\Drift Data No FFT SHIFT\008-RTto1000toRT 2xbin 20kx.wmv_driftdata.mat
Elapsed time is 197.272946 seconds.

Movie file 1

I allocated the new movie structure...
500 frames of 1999 done
1000 frames of 1999 done
1500 frames of 1999 done
I am writing the movie now...

Movie file 2

I allocated the new movie structure...
500 frames of 1999 done
1000 frames of 1999 done
1500 frames of 1999 done
I am writing the movie now...

Movie file 5

I allocated the new movie structure...
500 frames of 506 done
I am writing the movie now...
The corrected movies are complete ...enjoy...
Elapsed time is 102.879898 seconds.
fx >>
```

Figure 4: An example of the output displayed in the command window.

bar, it is straightforward to convert the shift values into length dimensions.

Another feature is that the drift correction is done in two steps. The first step calculates the amount of drift for each frame, and the second step corrects the drift and outputs a corrected movie file. It is possible to run one of these steps without running the other. For example, one could change the drift data and recalculate the output movie from the new
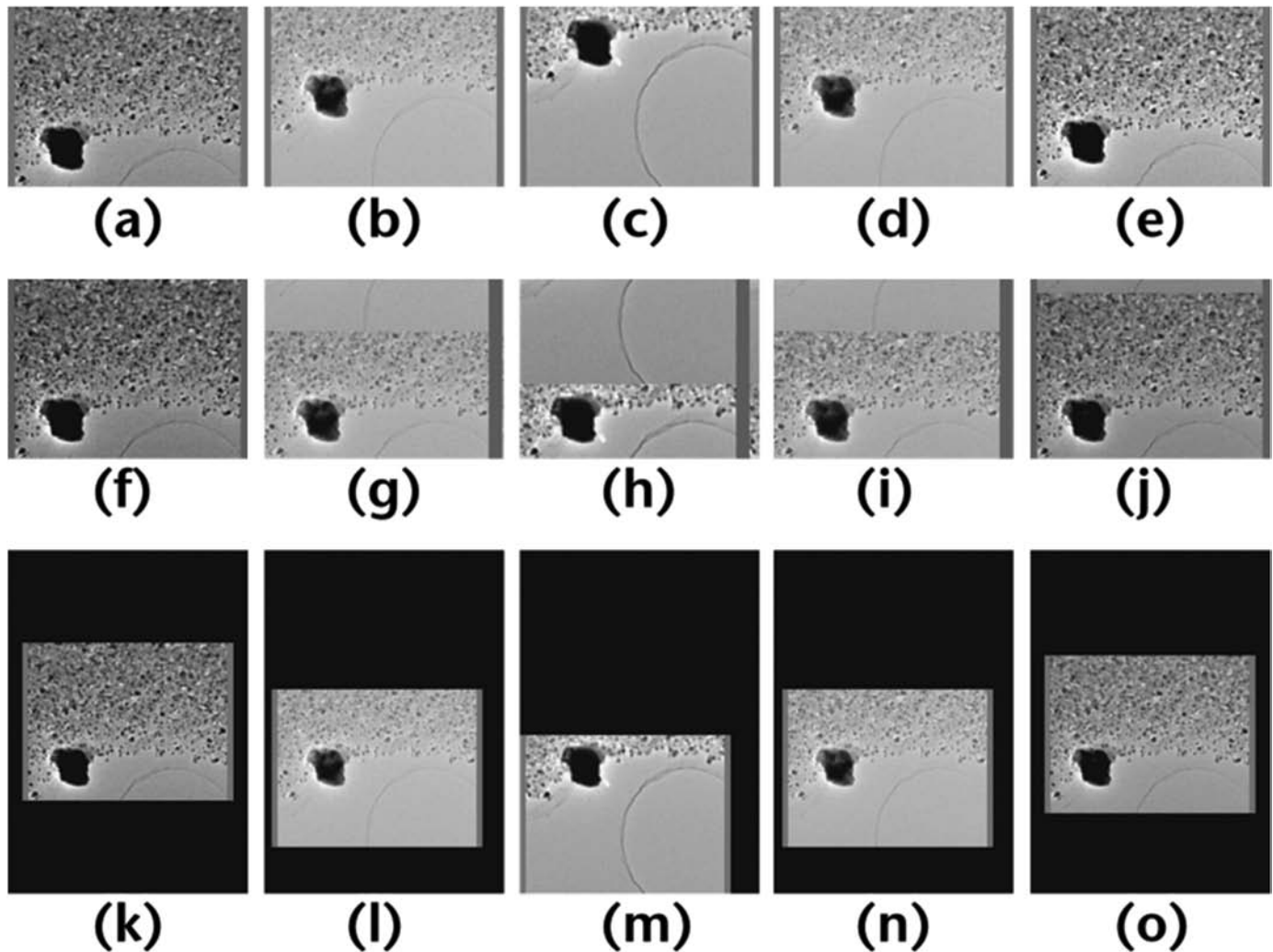
Figure 5: A series of images that demonstrate the difference between the "pad" and "wrap" option. The full width of the image frame is 3 μm. Raw image frames from a movie are shown in (a)–(e). The dark round features drift over the time frame of the experiment. The images in (f)–(j) show the results of the "wrap" option. The feature does not move from frame to frame, and the image size is equal to the images in (a)–(e). The images in (k)–(o) show the results of the "pad" option. Here, extra black space is added around the image frame so that the image can be shifted and the feature does not move.

drift data file. The second step starts on line 151 of the script. To run the first step only, place the cursor on a line in the script that is less than line 151 and hit ctrl+return (PC) or command+return (Mac). To run the second step only, place the cursor on a line greater than 145 and hit ctrl+return (PC) or command+return (Mac).

Depending on the computer configuration and the size of the input file, these calculations can take a long time. On a 2.4 GHz Intel i7 Macbook Pro running OSX 10.8.5 and Matlab 2013a, a 640 × 480 pixel movie with 20,538 frames took 2,018 seconds (34 minutes) to perform the drift calculation and another 3,363 seconds (56 minutes) to write the output movies. While the script is running, the command window gives information about the progress of the calculation. When using *drifty_shifty_deluxe.m*, it is recommended that the user check the command window every now and then to track progress. Figure 4 shows the kind of information that is displayed in the command window.

Finally, it may be desired to perform the drift calculation on only a selected area of an image frame rather than the

entire image frame. This can be done with a few changes to two lines of code. First, the reference image needs to be changed to just include the selected image region. This is on line 97, and the new code should read

$$\mathrm{imageref = frameref}(rowstart : rowstop, columnstart : columnstop, 1); \quad (6)$$

where *rowstart*, *rowstop*, *columnstart*, and *columnstop* are the matrix indices in the image that correspond to the region of interest. For example, if one wants to perform the drift calculation on only the boxed region of an image like Figure 6, then the appropriate values are *rowstart*=147, *rowstop*=347, *columnstart*=31, and *columnstop*=251. Then, a similar change must be made for each individual frame on line 111:

$$\mathrm{framei = imagei}(rowstart : rowstop, columnstart : columnstop, 1); \quad (7)$$

This change does not affect the output files. This means that the entire image frame will be corrected and included in the movie using the drift data calculated from the region specified above. If it is desired to extract only this region of interest for the output files too, then one should probably
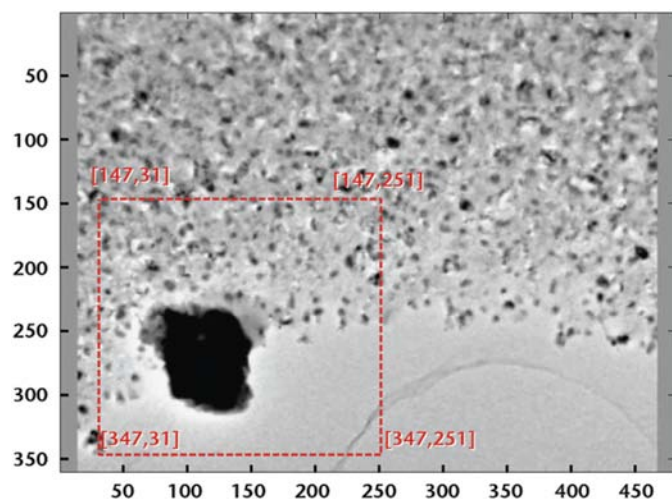
FIGURE 6: An illustration showing how to crop an image so that the drift calculation is only performed on a specified region of the sample. If one wants to perform the drift calculation on the boxed region of the image only, then one would use *rowstart*=147, *rowstop*=347, *columnstart*=31, and *columnstop*=251 to specify this in the drift calculation.

crop the movie before performing the drift calculation, or additional changes need to be made in line 245 and 253.

Conclusions

We have described and made available a Matlab script that can be used to correct spatial drift in movie data. This script is useful for *in situ* electron microscopy experiments, and it might also find useful applications in other areas. Our script is freely available and downloadable from the Matlab File Exchange Website, and it should work on multiple platforms. We have also described the general method of image alignment with Matlab functions so that the community can improve on our script if desired.

Acknowledgements

References

[1] J Damiano, DP Nackashi, and SE Mick, *Microsc Miroanal* 14 (2008) 1332–33.
[2] LF Allard, WC Bigelow, M Jose-Yacaman, DP Nackashi, J Damiano, and SE Mick, *Microsc Res Techniq* 72 (2009) 208–15.
[3] LF Allard, WC Bigelow, SA Bradley, and J Liu, *Microscopy Today* 17 (2009) 50–54.
[4] B Schaffer, W Grogger, and G Kothleitner, *Ultramicroscopy* 102 (2004) 27–36.
[5] JC Russ, *The image processing handbook*, CRC Press, Boca Raton, FL, 1994.
[6] http://www.mathworks.com/help/matlab/ref/videoreader class.html

— MT