

S. D. Barrett

Department of Physics and Surface Science Research Centre  
University of Liverpool, Liverpool L69 3BX, UK

Tel: +44-151-794-3894

Fax: +44-151-708-0662

Email: S.D.Barrett@liv.ac.uk

Continuing the theme of using software as a tool for microscopy, Steve Barrett discusses software applicable to the wide range of formats found in the spectrum of scanning microscopy techniques.

Steve Barrett is a Senior Lecturer in the Department of Physics and a Principal Scientist in the Surface Science Research Centre of the University of Liverpool. His research interests include the analysis of scanning probe microscopy images and rare-earth metal surfaces. He is the author of *Image SXM*, a software package written to handle the loading, display, processing and analysis of scanning microscopy images. He has authored over 50 papers on the application of ultraviolet photoelectron spectroscopy, low-energy electron diffraction and scanning tunnelling microscopy to metal surfaces and has co-authored a book on *The Structure of Rare-Earth Metal Surfaces*. His web page is at <http://www.liv.ac.uk/~sdb>

### Keywords:

Image analysis, image processing, digital images, software, scanning microscopy, file formats.

### Abstract

Digital images produced by scanning microscopes can be processed and analysed by software that is freely available. The problems associated with reading spatial calibration data and associated data acquisition parameters from image files using commercial software are outlined and solutions are presented. The flexibility of customisable software written to handle scanning microscopy images is illustrated using a number of features of the Macintosh application *Image SXM*.

### Introduction

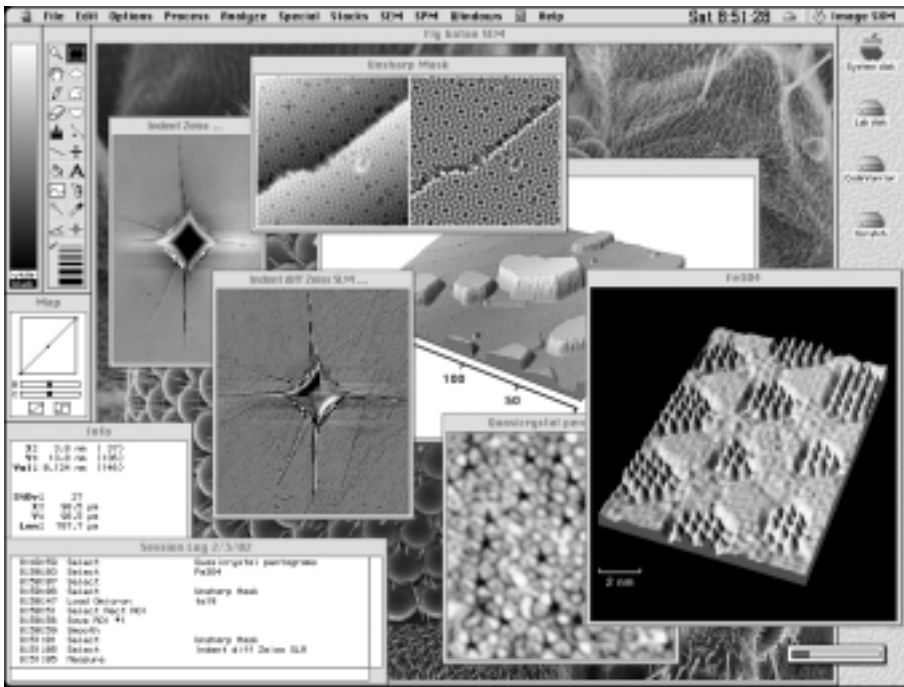
There are an increasing number of scanning microscopy techniques that produce digital images; some of the most popular are listed in Table 1. The images produced by these techniques share a number of related features: (i) the image files have a file format that is specific to the manufacturer of the microscope system, and (ii) the image files contain associated acquisition parameters, such as spatial calibration data for the x and y (and sometimes z) axes. If a user wishes to carry out off-line processing or analysis of their images using software other than that provided by the manufacturer of their microscope system, then these features can be a potential problem.

If any quantitative analysis of the images is to be carried out, then the spatial calibration of the image is of prime importance. Also, there may be many other data associated with an image, stored in the image files, that are of particular interest to a user. Some of these may be standard data acquisition parameters generated by the microscope system software when the image was acquired. Others may be comments entered by the user that are essential to the interpretation of the image content. To access the relevant parameters associated with an image the format of

**Table 1 Common acronyms in scanning microscopy**

SAM	Auger
SCM	Confocal
SEM	Electron
SFM	Force
SLM	Laser
SNOM	Near-Field Optical
SPM	Probe
STM	Tunnelling
SXM	Any of the above





**Fig. 1.** Screen dump of *Image SXM*, a version of the public domain image analysis software *NIH Image* that has been customised and extended for use with scanning microscopy images (<http://reg.ssci.liv.ac.uk>).

the data, and hence the details of the file format, must be considered.

Thus, there is a need for image processing and image analysis software that can be customised to handle image files from scanning microscopes. In this article, some of the software solutions that are available to users of scanning microscope systems are presented, and the relative merits of different methods of customisation are discussed.

### Image File Formats

The acquisition software that is provided by the manufacturer of a particular microscope system may have an option to save the images in a 'standard' file format (such as TIFF or JPEG), but this usually exists only to allow the user of the system to obtain images suitable for presentations or to insert into word processed documents. If this is all that is required by a user, then a number of different software applications are available, as reviewed by Entwistle (2001). However, if the user requires that the images maintain their spatial calibration, or if other data acquisition parameters are of importance, then many such applications are of limited use.

In most cases, when images are saved in a standard file format, the data acquisition parameters associated with the image are sacrificed. Only when the image is saved in the 'native' format of the manufacturer will all of the data acquisition parameters be saved along with the image. Even if the native format is TIFF (Tagged Image File Format), in which the parameters are saved as 'tags' in a well-defined format, many standard software applications that purport to read TIFF images will ignore all

but the most basic of the tags associated with an image.

Information concerning the file format is not always readily available, as manufacturers differ considerably in their willingness to reveal the details of the file format that they use. If the manufacturers are open about their file format, then determining how the image data and associated parameters can be read from an image file can be relatively straightforward. In some cases the parameters are stored as text, either in the header of the image file or in a separate parameter file. This allows them to be read by any application capable of displaying the contents of text files, such as word processors. This may be convenient for occasionally checking some of the parameters, but would quickly become tedious if it had to be carried out routinely every time an image needed to be spatially calibrated. If the parameters are stored in binary format in the header of the image file and the manufacturers are less forthcoming regarding the specifications of their file format, then a certain amount of detective work may be required to figure out how the data are stored.

### Software for Scanning Microscopy

There are a number of software packages available that are well-suited to handling scanning microscopy images and provide a reasonable range of options for image processing and quantitative image analysis. Those packages supplied by manufacturers of scanning microscope systems suffer from two main drawbacks. The first is the cost, which can be considerable for a large number of computers to be used independently from the data acquisition system. The second is the fact that the software will support only the image file formats of the manufacturer and not those of their competitors. For users who have many images produced on a variety of different scanning microscope systems, this can be a very inconvenient limitation.

A number of software packages are available at no cost to the user. These can be divided into two groups; those available in the public domain and those distributed by commercial companies. The principal difference between them is that for public domain software the source code that is used to create the applications is freely available to any user. The availability of source code is a benefit that cannot be over-emphasised. Even if the user chooses not to attempt to modify and recompile the code (the reasons for which will be dis-

cussed in the next section), then the ability to simply look at the source code to see what the program is actually doing is still very valuable. In some situations, the results of an image analysis routine can be of little value unless the user knows precisely the details of the algorithm employed to calculate the results.

Of those software packages in the public domain, one that has been available for many years and has built up a large number of users is *NIH Image* (<http://rsb.info.nih.gov/nih-image>), an Apple Macintosh application authored by Wayne Rasband of the National Institutes of Health in the USA. The original application was written as a general-purpose image analysis application intended for use with images obtained from light microscopes. As the source code of the application is in the public domain it has spawned many 'spin-off' applications over the years. One such application is *Image SXM* (<http://reg.ssci.liv.ac.uk>), a version written by this author that handles more than 20 different scanning microscopy image file formats and has many features intended specifically for use with SXM images. Details of how this customisation was implemented will be given in the next section, and examples of some of the features of *Image SXM* will follow later in the article.

*NIH Image* has been ported for use on Windows-based PCs by Scion, a manufacturer of video frame-grabbers, under the name *Scion Image* (<http://www.scion.com>). The source code for this PC version is not in the public domain, and so users of *Scion Image* do not have the same opportunities for customising the software as do users of *NIH Image*.

In an effort to provide image analysis software that is not specific to a particular computing platform, Wayne Rasband has been developing *Image/J* (<http://rsb.info.nih.gov/ij>), an application that provides most of the features of *NIH Image*. By writing the source code in Java, a platform-independent computing language, *Image/J* can be used on many different computers, including Macintosh, PC and Unix-based systems. At present, *Image/J* does not contain features specific to scanning microscope images, but the availability of the source code in the public domain means that such features may appear in due course as its usage grows.

In addition to applications based on *NIH Image*, there is another alternative for users of Windows-based PCs. Nanotec Electronica, a Spanish company that manufactures scanning probe microscopes, distribute an application called *WSxM* (<http://www.nanotec.es>) that currently supports the image file formats of their own SPM system plus two others; those of Digital Instruments and the Free University of Berlin. Future versions will support the image file formats of other manufacturers. As for other software supplied by commercial organisations, the source code is not available to users.

## Customising the Software

The degree to which any piece of software requires customising for use with scanning microscopy images depends on the requirements of the user. If a software package supports the particular image file format(s) of interest to a user, and if the processing and analysis features of the software include those that are needed by the user, then little or no customisation may be required. If some degree of customisation is required, then this can be implemented via one of three approaches: writing macros, using plug-ins, or modifying/extending the source code of the software. Each of these approaches to customisation will be discussed and the relative merits outlined.

### Macros

Macros are one of the simplest methods that allow customisation of image analysis software. *NIH Image* and its derivatives have a built-in interpreter that allows the execution of macros; short pieces of text that contain a sequence of commands. Macros, created using the basic text editor in *NIH Image* or a standard word processor, can be written by users without any programming experience. The commands in a macro often mirror those that are available via the menus of the application, so that a macro can be written that allows a sequence of operations to be applied to a file or an image (or a set of files/images), thus automating repetitive tasks (Barrett *et al.*, 1998, Bickmore *et al.*, 1999). This can include loading the image data from a file for display in a window and extracting data acquisition parameters to allow spatial calibration of the resultant image. Many example macros are supplied with *NIH Image* to demonstrate the ease of use and flexibility of this approach to customisation. Because of this combination of being in the public domain and being so flexible and extendible, *NIH Image* has built up a large group of users throughout the world. These users are supported by an email distribution system, or list, which keeps them all in touch with each other and allows for rapid exchange of information (instructions for subscribing to the *NIH Image* email distribution list can be found at <http://rsb.info.nih.gov/nih-image>). For instance, if a user has a specialist application for which a macro has been written then that macro can be placed onto the central NIH server, and a single email message will inform all users on the distribution list of its existence and availability.

By providing a short turnaround time between an initial idea and the implementation, testing and distribution of a macro to other users, this approach to customisation has meant that instances of users 'reinventing the wheel' have been much reduced. Macros have added support for specialist image file formats, and the content of some of the most popular macros have subsequently been incorporated into the code of later releases.

es of the application itself (the macro language is very similar to the Pascal language used for the *NIH Image* source code). By providing a means to 'prototype' an idea for a new feature and prove its general usefulness before it is integrated into the application, macros are an important element in the evolution of *NIH Image* and its derivatives. In addition, the contribution of macros to the ease and flexibility of user customisation means that they are responsible to a large degree for the popularity of *NIH Image* as a tool in image processing and image analysis. At present, macros cannot be used with the *WSxM* software.

### Plug-ins

The second approach to software customisation involves the use of plug-ins. Plug-ins are self-contained pieces of code that can extend the capability of a program. Depending on how they are written, they may be used with a variety of different programs. For instance, a plug-in written to acquire the data from a scanner for use with a commercial image processing program can also be used in *NIH Image* to do the same job. Plug-ins are a common feature of image processing software and many thousands are available from hundreds of commercial sources and via the internet as shareware or freeware downloads. They are less commonly used at present for supporting specialist file formats or for image analysis, and so their potential for use with scanning microscopy images has not been fully realised. This may be redressed by *Image/J*, in which plug-ins play an important part in the development and evolution of the software. All users of the application who need specialist functions are encouraged to write their own plug-ins and make them available to the rest of the user community, using a similar mechanism to that employed for user-written macros. At present, there are over 100 plug-ins available for download from the *Image/J* web site (<http://rsb.info.nih.gov/ij/plugins>), of which ~20 have been written to support specialist file formats and ~20 are categorised as image analysis. The remaining plug-ins are related to image processing or more general extensions to the application's utility. The *Image/J* web site also has links to other web sites from which plug-ins of interest to microscopists can be downloaded.

### Source Code

The third approach to software customisation is modification or extension of the source code, which is then put through a compiler to produce a customised variant of the original program. Although this requires some programming experience, the benefits can far outweigh the time invested in becoming familiar with the inner workings of the software. Macros, although easy to write and execute, cannot access all of the elements of the software that are available via the source code. By editing the source code, there is essentially no limit to the customisation that can be applied to file

handling, image processing and image analysis, or to the interaction via the user interface – the structure of the menus or the contents of dialog boxes, for instance.

If a user is trying to access the data acquisition parameters of an image file then a macro may offer an easy solution if the parameters are stored as text, which can be easily read and interpreted. However, if the parameters are stored as binary values, then the task of reading the data and interpreting the parameter values is made much easier if implemented in the source code. This is in part due to the fact that macros can use only a restricted set of data types to store the data read from files, compared to those available in the source code. In addition, there are often speed advantages to be gained by writing routines directly in the source code rather than in macros; the execution speed of some image processing or analysis algorithms can improve by more than an order of magnitude under favourable circumstances.

Modifying the source code of a program should not be regarded as a challenge requiring extensive programming experience. If the source code has been reasonably well written, then it is possible for a user to find those parts of the code responsible for producing a particular result and make small modifications to it. As familiarity with the source code grows with time, so more extensive modifications can be made. This was how *NIH Image* was extended by this author to create *Image SXM*. It started with modifications to provide support for a specific image file format and has continued to evolve, over the years, with the addition of image processing and image analysis features of interest to users of scanning microscopy systems. Also, support for more image file formats has developed by user demand. The following section gives more details of *Image SXM* as an illustration of how public domain software can be customised.

### Image SXM

The modifications to the source code of *NIH Image* that produced *Image SXM* can be divided into three categories: (i) support for different image file formats, (ii) extensions to provide additional image processing or image analysis routines, and (iii) refinements of the user interface.

In some cases, the addition of support for a new image file format required some detective work before it could be implemented. In others, the manufacturers of the scanning microscope system provided all the necessary information and so the coding was straightforward. One of the most important considerations was that the resultant application should be able to differentiate automatically between files generated by different microscope systems. This means that the user need not know the origin of a particular image file; the loading of the image and its spatial calibration are carried out

**Table 2**

The file formats of the image (or parameter) files supported by *Image SXM*. The first two columns identify the SXM data acquisition system. The four columns headed 'Parameters' indicate whether the image parameters are stored as text or binary values in the header or footer of the image file. No entry in either of the latter two columns indicates that the parameters are stored in a separate parameter file. The seven columns headed 'Image Data Format' indicate whether the image data are stored as integer or real data types, as 8-bit, 16-bit or 32-bit values, and as unsigned or signed values.

Manufacturer	System	Parameters				Image Data Format						
		Text	Bin	Head	Foot	Int	Real	8	16	32	+	+/-
Burleigh	ISTM		•	•	•	•			•		•	
Digital Instruments	NanoScope II	•		•		•			•			•
Digital Instruments	NanoScope III	•		•		•			•			•
DME	Rasterscope		•	•		•			•			•
Gatan	DigitalMicrograph	•			•	•			•			•
JEOL	SEM					•		•				•
JEOL	WinSem		•			•		•				•
JEOL	SPM		•	•		•			•			•
Klocke	Atomikro		•	•		•		•	•			•
Leica	TCS	•			•	•		•				•
LEO	SEM		•	•		•		•				•
Molecular Imaging	PicoScan	•		•		•			•			•
Noran	Vantage	•	•	•		•		•				•
Omicron	Pre-SCALA		•			•			•			•
Omicron	SCALA	•				•			•			•
Oxford Instruments	TOPSystem 3		•	•		•			•			•
Park Scientific Instr	HFS-LIF	•	•	•		•			•			•
Park Scientific Instr	HDF		•	•		•			•			•
Philips	SEM		•	•		•		•				•
Quesant Instruments	SPM		•	•	•	•			•			•
RHK Technology	SPM-32	•		•		•	•	•	•		•	•
TopoMetrix	SPMLab		•	•		•			•			•
Vacuum Generators	SAM		•	•		•				•		•
WA Technology	Pre-TOPSystem		•	•		•		•				•
WA Technology	TOPSystem 2		•	•		•			•			•
Zeiss	LSM		•	•		•		•				•

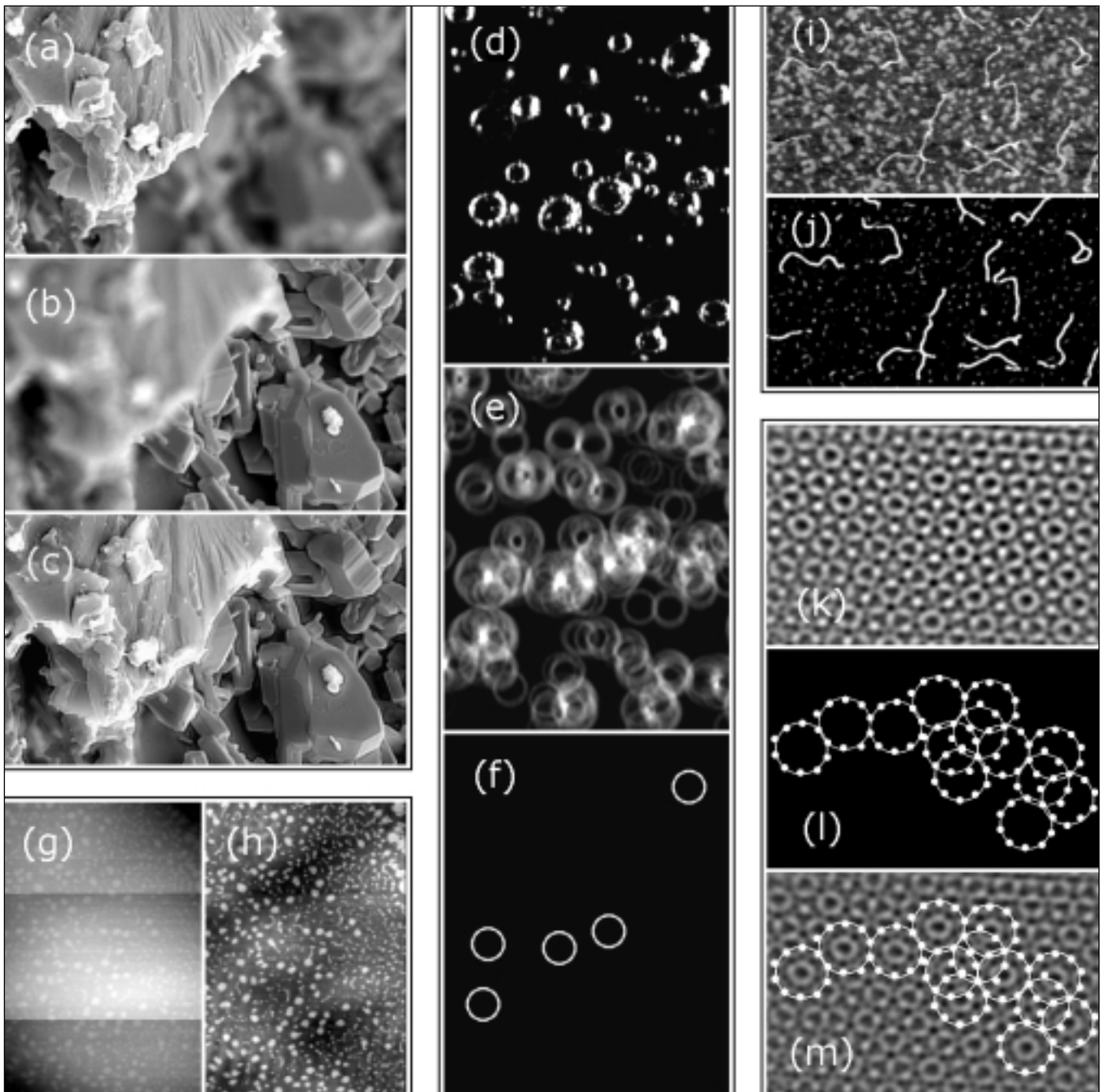
totally transparently to the user. *Image SXM* supports 26 image file formats related to scanning microscopy (see Table 2), plus eight other file formats, in addition to those formats supported by *NIH Image*. Although not immediately obvious from Table 2, the image file formats are all different from one another, as no two manufacturers have chosen to adopt the same format for storing image data and parameters.

The majority of the image processing or image analysis routines that have been added to *Image SXM* are principally related to scanning microscopy, though some have found more widespread use in other disciplines, such as satellite imagery. As the use of macros has played a pivotal role in the development and popularity of *NIH Image*, it was considered very important that macro commands were also added that would allow users full access to these routines. Thus, the 300 macro commands available in *NIH Image* have been complemented by 80 additional macro commands in *Image SXM*.

In addition to the substantial modifications noted above, other refinements have been added to *Image SXM*. For instance, the user-written comments that are associated with an image, which are read along with other parameters when an image is loaded, can be transferred to the comments field of the file. Thus, when the files are displayed by the Finder, the file handling component of the Macintosh operating system, they can be listed, sorted or copied according to the comments. Although a minor feature compared to image processing or image analysis routines, it is a good example of how access to the source code is sometimes the only option for customisation.

### Examples of Customised Features

A number of the features of *Image SXM* will be used to illustrate what can be achieved. They have been separated under headings of the particular scanning microscopy for which they were originally created, but



**Fig. 2.** Examples of image processing and image analysis routines that have been added to *Image SXM*.

(Top Left panel) Imaging a three-dimensional sample with an SEM will produce images with limited depth of field, throwing some parts of the image out of focus. A series of images focused at different depths (a, b) can be combined into a single image (c) by taking those parts of each image that are well-focused. SEM image provided by Sue Dipple, University of Birmingham, UK.

(Centre panel) Using a Hough transform for shape analysis: (d) Bubbles formed by the action of breaking ocean waves; (e) Hough transform for circles of radius 16 pixels; (f) Circles of radius 16 pixels centred at the points of highest intensity in the Hough transform. Thus the Hough transform has picked out all occurrences of circles of that radius in the image. The same process can be repeated for any other radii. Image provided by Dr Dale Stokes, Scripps Institution of Oceanography, University of California San Diego, USA.

(Bottom left panel) Removal of artefacts from scanned images can sometimes be achieved by treating each scan line of the image independently. Abrupt changes in the apparent height of a surface (g) caused by modification of the probe tip during the scan can be eliminated (h) by a line-by-line compensation algorithm. SFM image provided by Dr Dan Schwartz, Tulane University, New Orleans, USA.

(Top right panel) To extract quantitative information on the strands of DNA shown in the original image (i) it first had to be processed to separate the strands from the background topography (a rough surface). A customised algorithm was written to exaggerate the strands and suppress the background (j) so that the lengths of the strands could be calculated. SFM image provided by Dr Remus Dame, Leiden University, the Netherlands.

(Bottom right panel) Images of quasicrystal surfaces can be Fourier-filtered to make features easier to identify (k) and analysed with customised routines to detect ring structures with 10-fold symmetry (l). Superimposing the rings on the filtered image (m) allows the quasicrystalline features to be picked out. STM image provided by Dr Ronan McGrath, University of Liverpool, UK.

as most routines can be applied to quite diverse images this categorisation is somewhat arbitrary.

### *Scanning Electron Microscopy*

In light microscope and scanning electron microscopes (SEM) systems the sample may be non-planar or aligned at an angle to the optical axis of the microscope. In these circumstances, the restricted depth of field of the microscope can produce images that have different regions in and out of focus, as shown in Fig. 2(a-b). A series of images focused at different depths can be combined into a single well-focused image by taking an average of all of the images, with the relative weighting of each image calculated on a pixel-by-pixel basis (Fig. 2(c)).

Finding shapes in an image is a problem that is not specific to SXM images. The Hough transform is an image processing technique that can be used to pick out the outlines of shapes of a specific type and size, even if those shapes are not complete and are partly obscured by background clutter (illustrated in Fig. 2(d-f) using a light microscope image). The Hough transform has been implemented in *Image SXM* to pick out circles over a range of radii values and display a size distribution plot. This feature evolved from a user-written macro, but as it was written into the source code it has the advantage of significantly higher execution speed compared to the interpreted macro.

If a series of images has been acquired, either as a time sequence or as part of an experiment involving one or more parameters being varied systematically, then it is often desirable to accurately align each image such that identical (fixed) features occur at the same x-y coordinates in each image. This can be achieved at the microscope if great care is taken, or can be dealt with by manual correction of slightly misaligned images. To make this easier and faster, an autoregistration routine was written that compares each image in a set with a reference image (which can be any of the images) and makes the correction automatically. This is done using fast Fourier transforms (FFTs) to determine the extent of the misregistration between any two images. The inverse FFT of the product of the FFTs of two images, called the cross-correlation function, has a maximum value at the position corresponding to the misregistration of the two images. Thus, the cross-correlation of two images in perfect registry with each other would have a maximum at position (0, 0), whereas for two images displaced by five pixels horizontally and three pixels vertically, for example, the maximum would be at position (5, 3).

To mosaic a set of images into a larger composite image, the cross-correlation function can again be used. By comparing the edges of images that overlap, the extent of the overlap can be determined

and hence the images can be positioned at the appropriate position in the composite image. To eliminate the effect of abrupt changes in brightness or contrast at the seams of the composite image, an algorithm is used to weight the contributions of two, three or four overlapping images, producing a seamless mosaic.

### *Scanning Force Microscopy*

Artefacts can occur in scanned images for a number of reasons. For instance, if an abrupt change occurs during the scan that results in a jump in the recorded signal intensity, then the image will show a sharp discontinuity in the colour or shade of grey. As most scanning microscopes display images with the scan direction horizontal, this means that the image artefacts are horizontal lines that separate parts of the image with significantly different intensity, such as those visible in Fig. 2(g). When an image is loaded by *Image SXM*, the image data can be processed line-by-line so that the visibility of such artefacts can be reduced, or in favourable circumstances eliminated (Fig. 2(h)).

Many problems in image analysis can be reduced to the common problem of separating the desired features of the image from the background. In many cases, adjusting the brightness and contrast settings is not helpful as the background often has structure on a scale similar to that of the important features. The human eye/brain combination is very skilled at pattern recognition and is capable of picking out the required information ('signal') from an image full of distracting background clutter ('noise'). However, for quantitative analysis such an image needs to be processed to ensure that the background does not influence the results. An example of this is the analysis of strands of DNA shown in Fig. 2(i). A customised algorithm, which involves calculating the local curvature of the surface, was written to discriminate between the strands and the background features. In the resultant image (Fig. 2(j)) the strands of DNA stand out more prominently and their lengths can be analysed.

SFM systems can be used not only to provide topographic information about a surface, but also to measure the forces between tip and surface as a function of their separation. Although not image data, these force curves can help a user interpret their images, and so support for handling these files and displaying the curves was added to *Image SXM*.

### *Scanning Tunnelling Microscopy*

Quasicrystals are a class of materials that have long-range order but no periodicity. Unlike crystalline materials, they can have structures that exhibit local 5-fold and 10-fold rotational symmetry. Customised routines have been written to help analyse images of quasicrys-

tal surfaces, picking out features that have the required symmetry and displaying the results as an idealised model of the surface structure (Fig. 2(k-m)).

In addition to imaging a surface by mapping out the locus of a tip moving over a surface, maintaining constant tunnelling current for a given sample bias, an STM can also acquire spectroscopy data. Scanning tunnelling spectroscopy (STS) data comprise measurements of tunnelling current as a function of sample bias with the tip held in a fixed position. Depending on the STM acquisition software, the user can take a set of STS spectra corresponding to (i) every pixel in a topographic image, (ii) a grid of sample points, or (iii) a set of user-defined positions in the image. If STS data corresponding to a topographic image exists, then *Image SXM* allows the appropriate spectrum to be displayed and updated in real time as the user moves the mouse over the pixels of an image. This is another example of how modifications to the source code have allowed customisation that could not be implemented through the use of macros or plug-ins.

## Acknowledgements

The author thanks the following colleagues for kindly providing the images used to illustrate this article: Remus Dame, Sue Dipple, Ronan McGrath, Dan Schwartz and Dale Stokes.

## References

- Barrett S D, Bickmore B R, Rufe E, Hochella M F, Torzo G and Cerolini D (1998) The use of macros in AFM image analysis and image processing. *J. Comp. Ass. Microsc.* **10**, 77
- Bickmore B R, Rufe E, Barrett S D and Hochella M F (1999) Measuring discrete feature dimensions in atomic force microscopy images with *Image SXM*. *Geol. Mater. Res.* **1**, 5 (<http://gmr.minsocam.org/papers/v1/v1n5/v1n5abs.html>)
- Entwistle A (2001) Opening and viewing image data files. *Proc. Roy. Microsc. Soc.* **36**, 101.