The Use of Macros in Atomic Force Microscopy Image Analysis and Image Processing

Steve Barrett,[1,4] Barry R. Bickmore,[2] Eric Rufe,[2] Michael F. Hochella,[2] Giacomo Torzo,[3] and Dino Cerolini[3]

The public domain image analysis program *NIH Image* (http://rsb.info.nih.gov/nih-image) has been modified and extended to produce *Image SXM* (http://reg.ssci.liv.ac.uk), a program that supports many of the image file formats used by various scanning probe microscope manufacturers. This article discusses the different approaches to software customization and describes the use of two sets of macros in the analysis and processing of atomic force microscopy images.

KEY WORDS: Image analysis; image processing; customization; atomic force microscopy macros.

APPROACHES TO CUSTOMIZING IMAGE ANALYSIS SOFTWARE

The public domain image processing and analysis program *NIH Image* (http://rsb.info.nih.gov/nih-image) has been used with scanning probe microscope (SPM) images for many years and has evolved into one of the most powerful and flexible programs of its type. Although it was written as a general-purpose image analysis program, its popularity is the result of its ability to be extended and customised to suit the specific needs of the user. The functionality of *NIH Image* can be extended using two different approaches —writing macros and modifying the source code. These are complementary in the sense that neither approach on its awn offers the user access to the full power and flexibility of the program.

A macro is a set of instructions executed by an interpreter integrated into *NIH Image*. Each instruc-tion is interpreted every time it is executed, in contrast to the operation of a compiler that converts all of the Pascal code of the program (source code) into the resultant application (object code) before the latter is executed. Although this results in interpreted code being intrinsically slower in execution than compiled code, macros have a number of advantages that make this approach to customization popular among many users. *NIH Image* supports hundreds of macro com-mands that give users access to essentially all of the image handling, processing, and analysis features of the program—any action that can be invoked via the interactive menu-driven user interface can alter-natively be executed as a macro command or short sequence of commands. One popular use of macros is to automate simple but repetitive sequences of pro-cessing or analysis procedures for batches of images. More complex macros can combine commands to construct customized image analysis routines for spe-cialized applications. They can be easily run, modified, and run again, resulting in a fast "turn around" time between an initial idea and the implementation of a working solution. The macros are written in a Pascal-like language and are saved as text files, facilitating the exchange of macros between users. For those users who need to write macros only occasionally, and are hence less familiar with the structures of macros and the commands available, the readability of macro text

files is important as this allows macros written by other users to be adapted for their needs.

The alternative to writing macros is to modify the source code. As the source code of *NIH Image* is in the public domain, it can be modified and recompiled to produce a customised version of the program. This approach makes it possible to integrate new features seamlessly into the existing infrastructure of the program. The programmer retains full control over the user interface, the coding of the new procedures or functions and their interaction with existing features of the program. Although this degree of control is highly desirable, some programming experience is required to make this approach viable.

One such customized version of *NIH Image*, called *Image SXM* (http://reg.ssci.liv.ac.uk), has been created specifically to handle images from scanning microscope systems. It supports SPM images from many different manufacturers (including Digital Instruments, JEOL, Omicron, Oxford Instruments, Park Scientific Instruments, RHK Technology, Topometrix, and WA Technology), each having a unique file format for the images and associated acquisition parameters. Support for a particular SPM image file format requires knowledge of where the image data is located, the type of data that is used to represent each pixel value (e.g., 16-bit integers, 32-bit reals, etc.), and the location and format of the acquisition parameters that can be used to determine the calibration of the xy and z scales of the image. Extraction of other parameters from the image files is also desirable, but the details will differ between different file formats and according to the needs of individual users. By extending the image loading routines of *NIH Image* to support SPM images, *Image SXM* allows SPM images to be loaded using the same menu and macro commands as for images in standard formats (such as TIFF or PICT), thus making the process essentially transparent to the user. Although it is possible to add support for SPM image file formats by writing macros, this level of integration can only be achieved through modification of the source code.

In addition to support for SPM image file formats, *Image SXM* has many SPM-specific features that have been requested by users. Some have been an implementation in source code of a procedure that had been written originally as a macro, but that had been found to execute at an unacceptably slow speed. Others have been added to provide the functionality of commercially available software. The source code of *Image SXM* has been written without compromising any of the functionality of the macro programming

that has made *NIH Image* so popular. Indeed, the procedures and functions that have been added to *Image SXM* are accessible from macros through more than fifty new macro commands and functions, thus encouraging the continued use of macros in user customization. In the following sections, users of *Image SXM* describe macros written to analyze and process AFM images.

ATOMIC FORCE MICROSCOPY IMAGE ANALYSIS

Measuring the Perimeter, Area, and Volume of Discrete Features

One of the most useful aspects of SPM is its ability to quantitatively measure the spatial dimensions of surface features such as etch pits, as well as small particles deposited on a substrate. Some features, such as euhedral etch pits, are conveniently characterized using the standard analysis software that comes with SPM systems, but others have complicated shapes, and require more sophisticated analysis. We have created a suite of *Image SXM* macros to measure the perimeter, horizontal area, and volume of discrete surface features and particles in atomic force microscopy (AFM) images, obtaining accurate and consistent estimates [http://www. geocities.com/Athens/Parthenon/2671/Macros.html; complete directions for using this macro suite are available in Bickmore *et al.*, 1999]. One aspect of *Image SXM* that makes it especially convenient for the measurement of irregularly shaped features is that it allows the user to utilize rectangular, oval, polygonal, and freehand selection tools to define a region of interest (ROI) of any shape, thus excluding extraneous features. This powerful feature is simply not available in the commercial SPM analysis software with which we are familiar.

Perimeter and Horizontal Area Algorithms

It is a simple matter to measure the perimeter and horizontal area of a discrete feature in a binarized image, and *Image SXM* has built in tools to calculate such values. The difficulty in such calculations, however, is deciding at which height level to set the threshold, as the effect of altering the threshold height even slightly can be drastic, especially in the case of irregularly shaped features. In order to estimate consistent values from feature to feature and from image to

image, one must decide upon some optimum threshold level at which to make the measurements. Therefore, our *Image SXM* macros threshold the image at 254 of the 256 possible gray levels (excluding white and black), and calculate the perimeter and area of the particles in the selected area at each setting. The perimeter vs. threshold height curve is then subjected to a 3-, 5-, 7-, or 9-point (user-defined) smoothing routine, and the derivative of the perimeter vs. threshold height curve is calculated at each height level from the smoothed curve. The resulting derivative curve is essentially a map of image complexity vs. threshold setting. That is, perimeter values where the feature boundaries are the least complex will change the least from threshold level to threshold level, and so will produce values near zero in the derivative curve (Russ, 1990). Optimum threshold levels are selected where the absolute values of a string of five or more consecutive derivative points fall below a user-defined tolerance level. Even where this algorithm cannot select an optimum height at which to measure these feature dimensions, the user can plot curves for perimeter, area, and perimeter derivative vs. the threshold height using separate macros, and select a threshold height manually.

An example of the utility of these routines is shown in Fig. 1. Figure 1(a) shows an AFM image of a montmorillonite clay particle, along with a polygonal ROI selected just around the boundary of the particle. In Fig. 1(b) a cross section of this particle is shown, with dashed lines indicating the levels the computer picked as optimum threshold heights. It can be seen that the algorithm picks the optimum height levels to minimize both the effect of edge broadening due to tip geometry and the effect of irregularities at the very top surfaces of the particle.

Volume Algorithm

The volume of a feature can easily be calculated by first defining a baseline height, and then multiplying the area of the selected ROI by the average pixel height (relative to the baseline height) within the ROI (Russ, 1995). This method of volume calculation assumes implicitly that any noise in the image is distributed symmetrically about the true height values. It also ignores effect of AFM tip geometry and other artifacts on the perceived volume, although a user could modify the macro code to account for such effects for specific applications (the effect of tip geometry is discussed in the next section).

The volume calculation macros define the baseline height by first creating a histogram of the number of pixels at each height level within the ROI. The histogram is then subjected to a 5-point smoothing routine, after which the height level of the baseline is determined by identifying the maximum of the first large peak from the bottom (in the case of particles) or top (in the case of pits). If there is a significant deviation from the overall baseline in an area within the ROI, a smaller peak in the histogram might interfere with the baseline determination. Therefore, the user is asked to define a minimum (particles) or maximum (pits) height level for the baseline to exclude such anomalies. Another macro allows the user to plot the smoothed histogram, against which the computer-selected baseline height, as well as the shape of the background noise distribution, can be checked. If an anomalous histogram peak interferes with the selection of the baseline, the volume macro may be run again, and a minimum or maximum baseline height may be entered to exclude a false result.

ATOMIC FORCE MICROSCOPY IMAGE PROCESSING

Correcting for Tip Locus Effect

When a surface is imaged using AFM the acquired set of data values (z height as a function of xy position) does not represent the true surface topography. As the tip has a finite width, and usually an irregular shape, the data are the result of the surface topography convoluted with the tip geometry. Here, the "convolution" is not a mathematical convolution of two functions, but a geometric effect of the requirement that the tip remains in contact with, or at a fixed height above, the surface. The effect of the tip geometry on the observed profile of a surface feature is illustrated in Fig. 2. Partial compensation for the distortion of the AFM profile may be obtained using the macro "Erosion," also known as "Tip Locus Effect" (http://www.geocities.com/Athens/Parthenon/2671/Macros.html).

The macro works by calculating the highest surface that is consistent with the observed AFM profile. For this to work, the geometry of the tip used to obtain the AFM image must be known. The tip profile is mapped on to the image with the center of the tip positioned at a particular pixel. If any part of the tip is lower than the AFM profile at the corresponding pixel, then this indicates that the true surface must be
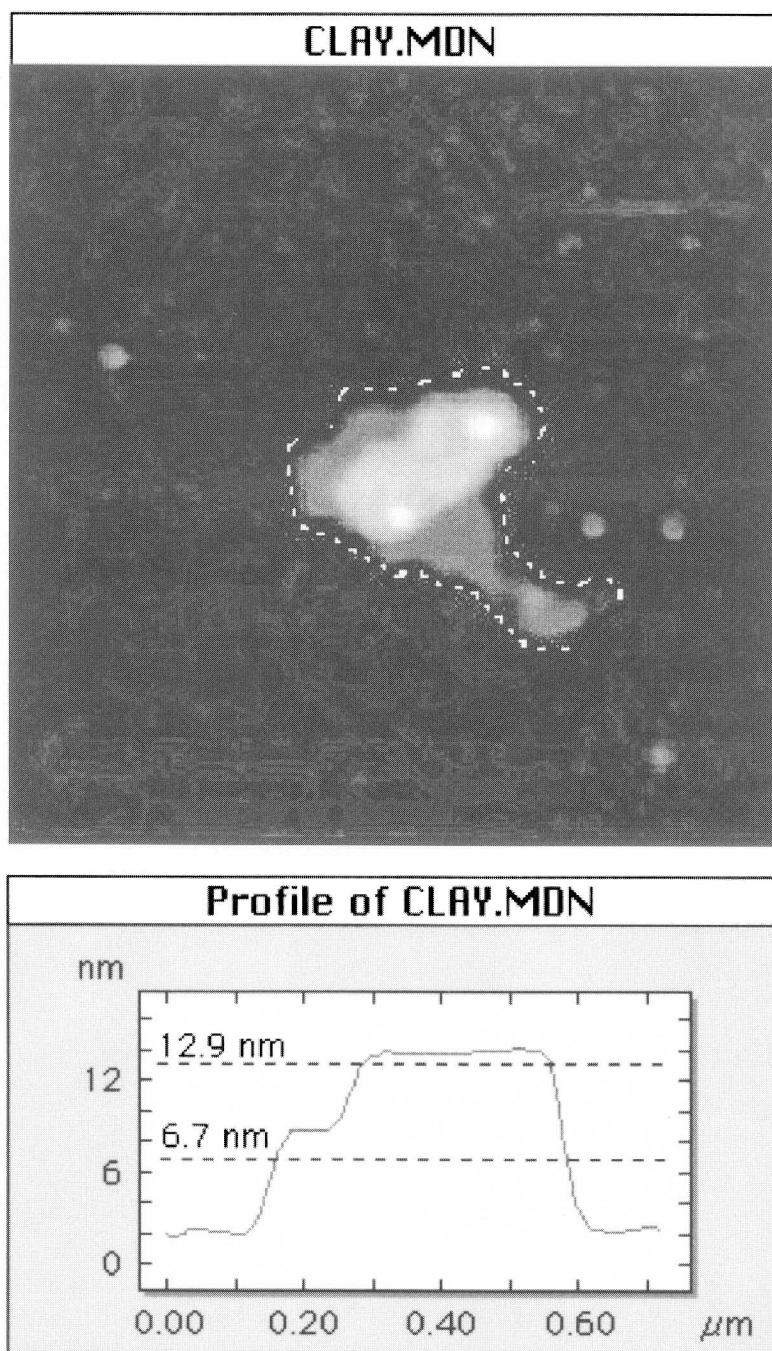
Fig. 1. The $1.7 \times 1.7\ \mu$m AFM image of a montmorillonite clay particle under deionized water. The particle has well-defined terraces at ~ 8 and ~ 14 nm height, with the baseline at ~ 2 nm. The dashed line in (a) represents the ROI used for the measurement routines. The dashed lines in (b) indicate the threshold levels at which the perimeter and area were measured. Both (a) and (b) are digital captures of windows in *Image SXM*, taken during the measurement procedure, with the dashed lines in (b) being added afterward.
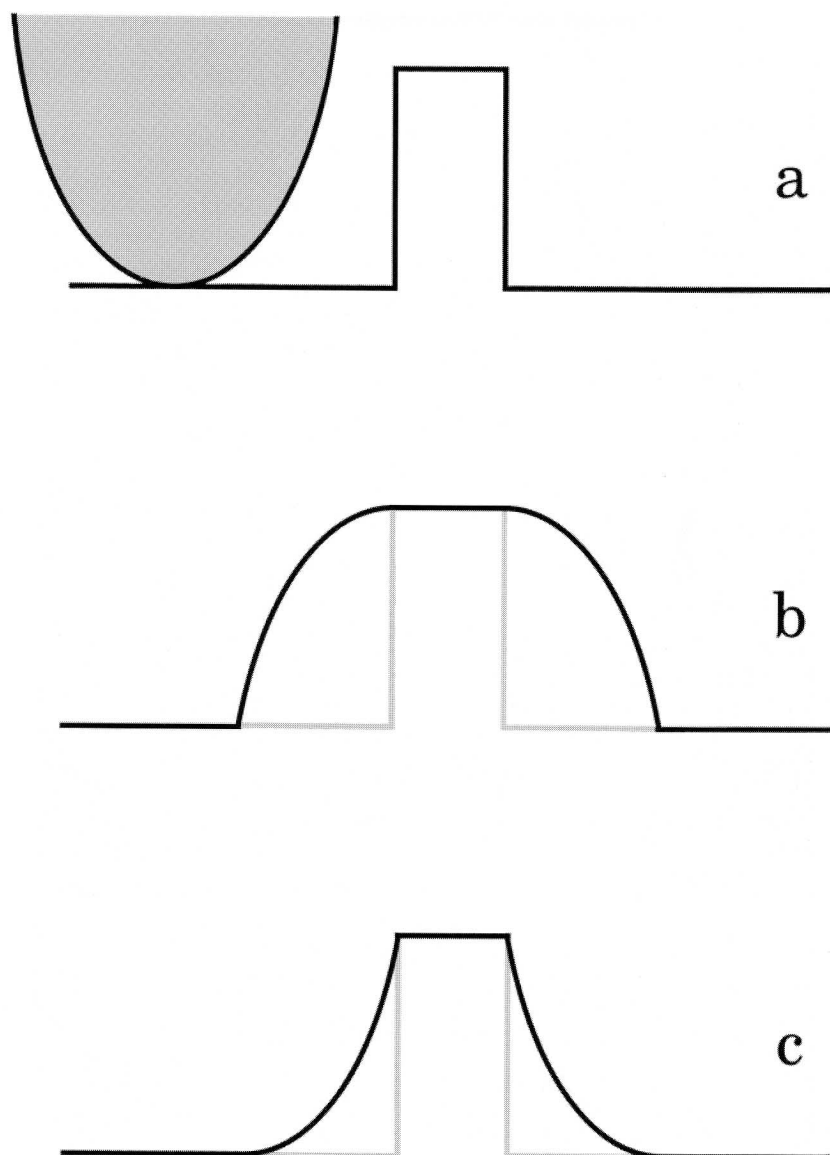
Fig. 2. Compensation for the effect of a sharp surface feature being imaged by a relatively blunt tip. (a) The profile of an idealized feature to be imaged by the parablolic tip shown in grey. (b) The locus of the lowest point of the tip assuming that the tip and the surface stay in contact throughout the scan. Note that if the feature is an infinitely narrow spike, then the locus of the tip would be an inverted profile of the tip. (c) The result of applying the "Erosion" or "Tip Locus Effect" routine to the profile shown in (b). Note that although the surface height close to the feature cannot be calculated because of the finite tip width, profile (c) is closer than profile (b) to the true profile of the feature shown in (a).

lower than that indicated by the AFM profile, which is adjusted accordingly. This process is then repeated, such that the tip is mapped over every pixel of the image. When the process is complete, the adjusted AFM profile represents the highest surface that would give the original AFM profile if scanned using the specified tip profile—the macro cannot tell if the true surface has some areas that are lower, as these could be "shadowed" by the scan of the finite-width tip over relatively sharp surface features.

In addition to correcting for the tip locus effect of images, a similar algorithm can be used to correct the

one-dimensional data of line profiles taken across images. The macro needs to be able to redraw the plot of the line profile after it had been modified by the erosion algorithm, but this was not a feature that was anticipated in *NIH Image*. As a result, the macro command "ReDrawPlot" was added to *Image SXM*. This is another example of the complementarity of the two approaches to customization—source code and macros —where each can extend the functionality of the program in different ways.

REFERENCES

(http://rsb.info.nih.gov/nih-image).

(http://reg.ssci.liv.ac.uk).

Bickmore, B. R., Rufe, E., Barrett, S. D., and Hochella, M. F., Jr. (1999). Measuring discrete feature dimensions with Image SXM. *Geol. Materials Res.*, submitted.

Russ, J. C. (1990). *Computer-Assisted Microscopy: The Measurement and Analysis of Images*, Plenum, New York, pp. 108–115.

Russ, J. C. (1995). *The Image Processing Handbook* (2nd ed.), CRC, London, p. 541.

(http://www.geocities.com/Athens/Parthenon/2671/Macros.html).