Efficient Processing of Data Acquired Using Microscopy Techniques

Ştefan ȚĂLU1,*, Dinara SOBOLA2, Nikola PAPEZ2, Rashid DALLAEV2 and Petr SEDLÁK2

1Technical University of Cluj-Napoca, The Directorate of Research, Development and Innovation Management (DMCDI), Constantin Daicoviciu Street, No. 15, Cluj-Napoca, 400020, Cluj County, Romania

2Brno University of Technology, Faculty of Electrical Engineering and Communication, Physics Department, Technická 8, 616 00 Brno, Czech Republic

*Corresponding author

Keywords: Data processing, MATLAB programming language, Microscopy, Topography.

Abstract. This article focuses on processing of data acquired by the atomic force microscopy (AFM) method using tools of MATLAB programming language. The basic examples are presented as codes for two-dimensional (2D) and three-dimensional (3D) images of surface microstructure, heights histograms and S parameters. Furthermore, these examples can be applied to other scanning probe microscopy methods. The advantage of such processing over the software which is supplied to the microscope is complete freedom to work with the obtained data.

Introduction

An important aspect in the creation and writing of articles, presentations, posters and other scientific publications is their presentation, stylistics and appearance. A scientific work with high regards to the aforementioned factors is more likely to capture a potential reader. Particularly, when publishing data in applied physics and other material sciences where we encounter many results in the form of graphs and images of surface structures that form the core of the work. It is known that sometimes graphs and images happen to be unreadable, either because of rasterization of an image, which causes deterioration in its quality or small fonts. The solution in this case is to vectorize the image or modify its various parameters to improve readability. Unfortunately, not every software supplied to measuring devices allows such a wide range of changes in the image or graph properties, and then an author will depend on the limited capabilities of the original measuring program. One solution to effectively manipulate measurement data is to process it using MATLAB software for engineers and scientists. This well-known software designed for both research and students allows for almost unlimited changes in visualization as well as adjustments of measured results.

AFM technique is a very high-resolution type of scanning probe microscopy (SPM) that is capable of providing information on surface texturization at micro- and nano-scale [1-8]. The 3D model of the sample surface is reconstructed from individual positions of the tip tool [9-12].

The following work describes the processing and visualization of the surface structure scanned using atomic force microscopy. Since the AFM microscope can scan a structure up to atomic dimensions, this morphology analysis is very often used in scientific studies. However, their results are not always correctly interpreted.

Description of Basic Data Processing

To obtain the data from the sample, the SPM microscope NTEGRA Prima in semi-contact AFM mode and its software Nova v.1.1.1.16703 have been used. For the data processing we used MATLAB 2017b, therefore steps described here are demonstrated with this version. However, the method of data processing mentioned here can be implemented with minor modifications using other measurement software or another version of MATLAB. The only condition is that the microscope software must be capable of exporting the data in a plain text format, such as a *.txt, *.csv or *.m file.
Our data describes the properties of a 10×10 μm structure. These dimensions are presented by 256 points in both x-axis and y-axis. Each point carries surface height information and all points put together form a matrix [13].

The first step is importing data to MATLAB. You can either write your own import script or let MATLAB generate the script itself - the Home tab, the Import Data button - here, define the matrix data, then for importing select the Import tab and then the Import Selection - Generate Script button. This generated script can then be inserted into the main data visualization file. And finally, fastest yet least effective option is to manually copy the data from the exported file directly to MATLAB.

**Visualization of Surface Structure Contours in 2D Views**

The first example can be a depiction of contours on the sample surface. Each color in this case represents a certain height of the structure after a certain step of your own choice. Similarly, as with the import of measured data, we can use the MATLAB graphical interface to select a graph. At first you need to load your own variable with measured data. It is selected in the **Workspace** window and under the **Plots - contour** tab a required graph type can be chosen. The open window already shows the resulting graph. Then we can manually edit individual elements using the **Plot Tools**, which is accessible from the **Edit - Figure Properties** menu. A graph can also be selected using the Command Window by typing: `contour(measured_data)`.

In Figure 1, contours are visualized using two types of functions. In Figure 1a with `contourf` function, contours are clearly defined by a black line. Figure 1b is a different variation, where the black contours are no longer visible and the color theme has been changed as well as the step defining the height level. The images are not exactly identical because another function `contour` has been used. The 2D map image in Figure 1a is therefore particularly useful in the field of study where we need to emphasize the exact outline of the peaks. If we want to emphasize the overall structure rather than concentrate so much on outline contours, it is possible to use less contrasting solution as shown in Figure 1b [14].

![Figure 1. 2D image of morphology using functions a) contourf and b) contour.](image)

**Visualization of 3D Morphology and Its Filtering**

If it is necessary to visualize a sample surface in the 3D view, as shown in Figure 2 it is selected the **surf** graph type. Also, if an image is too noisy, a filter can be applied before the data is displayed. The filter can be customized to simpler types. In our case, in Figure 2b, the data was filtered using Gaussian smoothing with standard deviation 3, which can be done with the `imgaussfilt3` function. The study of the filtered surface in the space is considerably less complicated and it is easier to identify the individual projections and to form a better idea of its structure [15].
Combining 3D and 2D Views with Other Methods

The ideal form of data processing would be the combination of the two previous methods, where the surface contours in 2D images mirror the surface of 3D graph morphology. The surface phase can also be obtained using the AFM microscope, so the 3D image can also mirror its phase in 2D processing [16].

Using the `colormap` command we can define a color theme, as can be seen from Figures 1a and 1b. The most important is the `surf` command and the `contour` command. Here the variable `measured_data` specifies the input data that were taken from the microscope. These data can be defined either as a three-dimensional matrix or 3 vectors representing the x, y and z axes. The `LevelStep` parameter then determines the step after which the color changes in the graph. In our case, therefore, each contour is represented after a 7 nm step. In order for the image to create a better impression in 3D space, it is necessary to illuminate it. `View` parameter can also be used to rotate the graph. The definition of special characters, such as "î", can be written according to the ASCII table [17-19].
Other Possibilities for Processing Morphological Data

A plethora of data visualization variations of surface morphology is not the only option. From the measured surface data we can also evaluate so-called $S$ parameters, which are parameters indicating significant morphological properties of materials. Besides the maximum peak height $S_p$ or the whole structure $S_z$ including negative values, it is possible to perform a distortion analysis of the roughness shape $S_{sk}$, the sharpness of the peaks $S_{ku}$ or for instance the number of peaks $S_{pd}$. As an example, $S_p$ can be calculated by: $\max(\text{measured\_data}(\cdot))$. Besides the $S$ parameters, it is also possible to create a histogram of heights with another simple command: $\text{histfit(reshape(measured\_data,\[],1))}$. The whole graph can, of course, be modified arbitrarily and in detail [20].

![Height histogram for Figure 3.](image)

Summary

Using the above mentioned methods, MATLAB can simply visualize and almost arbitrarily modify surface morphology at a very good level without any limitations intrinsic to other software. As a relatively practical function for visualizing surface morphology, $\text{surf}$ and $\text{contours}$ are mentioned. The combination of 3D and 2D images gives a comprehensive overview of the structure. The resulting graphs can then be exported to a wide variety of image formats, including vector ones. It should be noted, however, that in the case of the vectorization of an image, it is also necessary to calculate its resulting bit size and computer processing time. While Figure 1 can be easily inserted into the publications, Figure 3 may complicate its processing time as it contains a large number of vectors. The solution is rasterization at the expense of poor imaging quality.

Acknowledgement

Research described in this paper was financially supported by the Ministry of Education, Youth and Sports of the Czech Republic under the project CEITEC 2020 (LQ1601), by the National Sustainability Program under grant LO1401, by Internal Grant Agency of Brno University of Technology, grant No. FEKT-S-17-4626. The research infrastructure of the SIX Center was used in this work. Part of the work was carried out with the support of CEITEC Nano Research Infrastructure (ID LM2015041, MEYS CR, 2016–2019), CEITEC Brno University of Technology.

References


