Visualizing and Analyzing 3D AFM data with 3ddatadiver

Renlong Zheng, Xueqiao Zhang, Ellen Murphy
Department of Materials Science & Engineering
University of Washington, Seattle, WA

https://github.com/EdwardZheng0312/3ddatadiver

Introduction & Background Information

Recent developments in atomic force microscopy (AFM) technology allow us to directly image hydration layers at solid-liquid interfaces using 3D mapping. **3ddatadiver** allows AFM instrument users to process and visualize 3D data. The important steps are data processing, visualization, and processed data file exports. So far there is no other software available that allows for 3D AFM data exploration. This project seeks to develop a GUI through which users can interact with the data by viewing a full 3D rendering and slices in 3D or 2D cartesian coordinate systems.

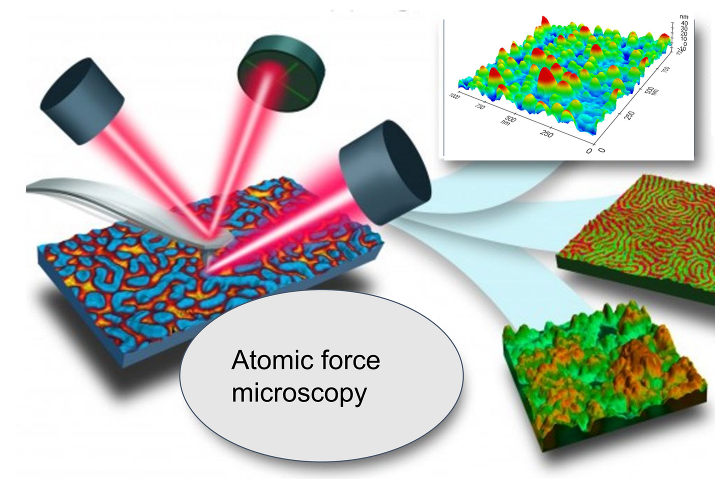


Figure 1: Working of atomic force microscopy[1]

AFM is a type of scanning probe microscopy (SPM), with demonstrated resolution on the order of fractions of a nanometer, more than 1000 times better than the optical diffraction limit. The information is gathered by "feeling" or "touching" the surface with a mechanical probe[2].

Data Cleaning

In addition to standard 2D AFM data cleaning techniques, there are unique problems presented in cleaning 3D AFM data. With both 2D and 3D AFM data processing the slope of the sample must be corrected as the sample stage may not be flat, the analysis may be implemented at a region with a local slope that is not representative of the global sample shape, and surface hills and valleys must be accounted for.

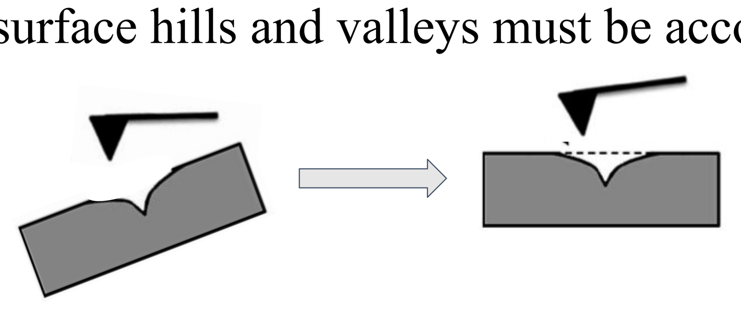


Figure 2 : Cartoon of surface correction calculations. The dotted line in the image on the right is the mean average surface value.

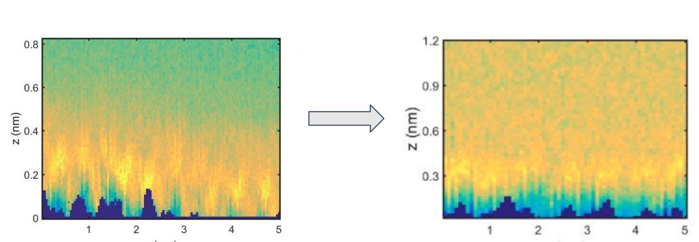


Figure 3 : Images generated from real 3D AFM data. Preprocessed data (left) and after slope correction (right).[3]

The raw data yielded from 3D AFM analysis is a 3D array, which emulates x, y, and z cartesian coordinates populated with a signal value (Amplitude, Phase, Drive, ect..) at each coordinate. The challenge with 3D AFM data is to take the corrections applied using traditional AFM data cleaning methods and extrapolate them to the 3D dataset in a way that preserves the true position of the each data point. This is achieved by generating a Z-vector that is drawn from the top of the sample to a point slightly below the calculated surface of the material and aligning the data at each x,y coordinate with the same Z-vector.

Data Visualization

Data cleaning yields a 3D numpy array which the user can visualize using a 3D plot or extract slices parallel to X, Y, or Z axis in cartesian coordinate systems.

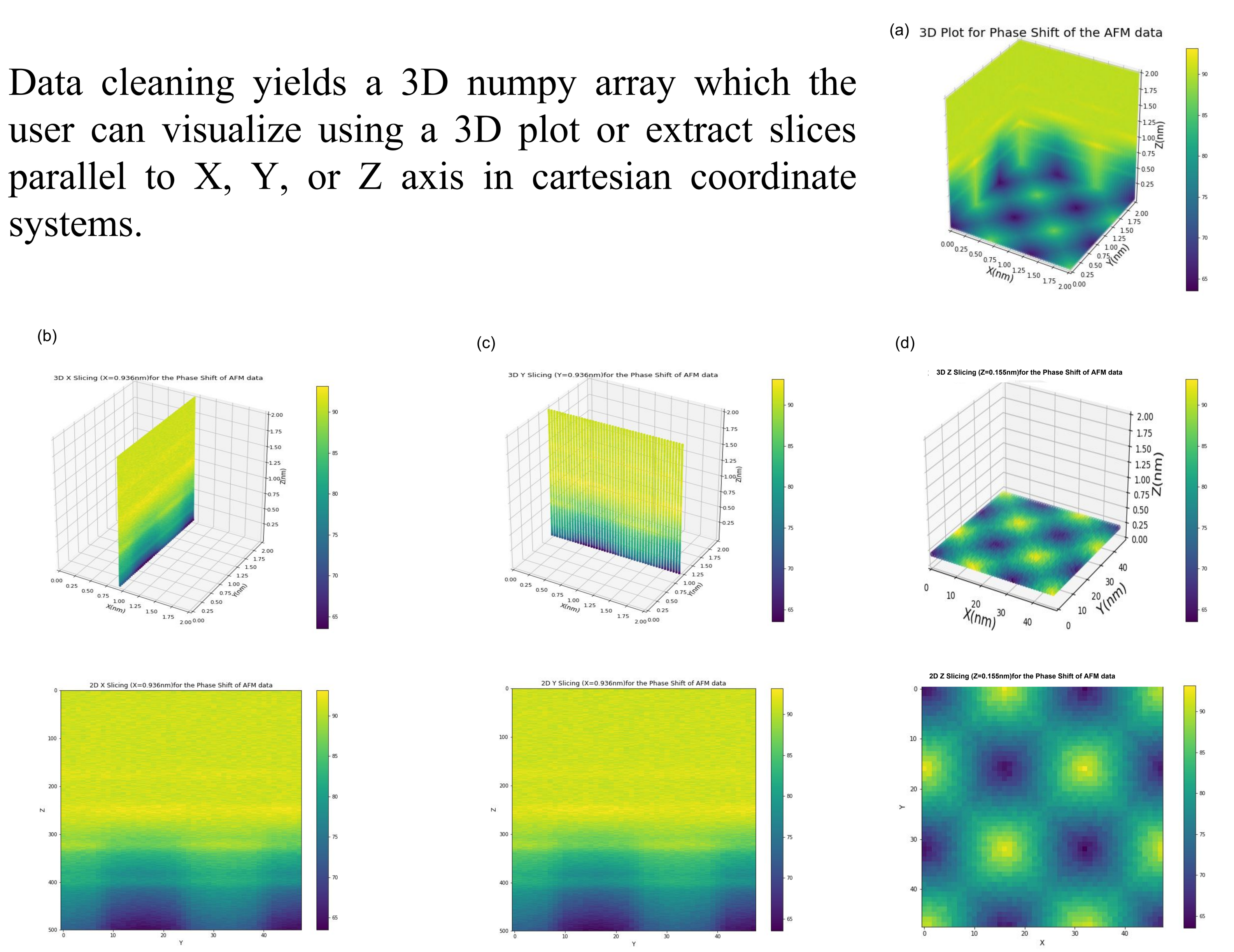


Figure 4: (a)3D plot; (b)3D slicing(X axis) and 2D slicing(X axis) (c) 3D slicing(Y axis) and 2D slicing(Y axis) (d)3D slicing(Z axis) and 2D slicing(Z axis) for AFM phase shift

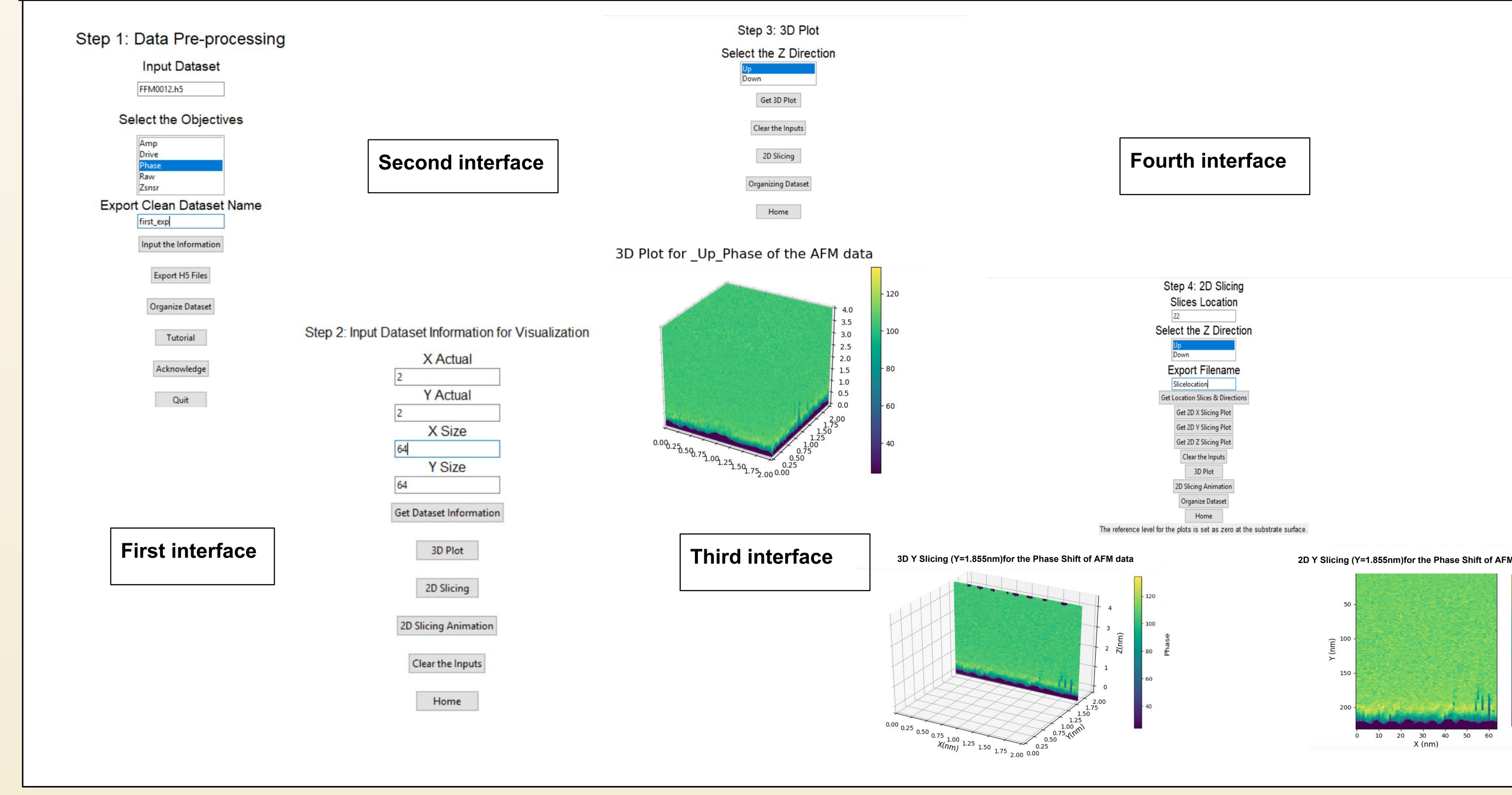
GUI Design

In order to help users of all backgrounds interact with AFM data through visual indicators instead of complicated code, we built a graphical user interface (**GUI**) with **Tkinter**.

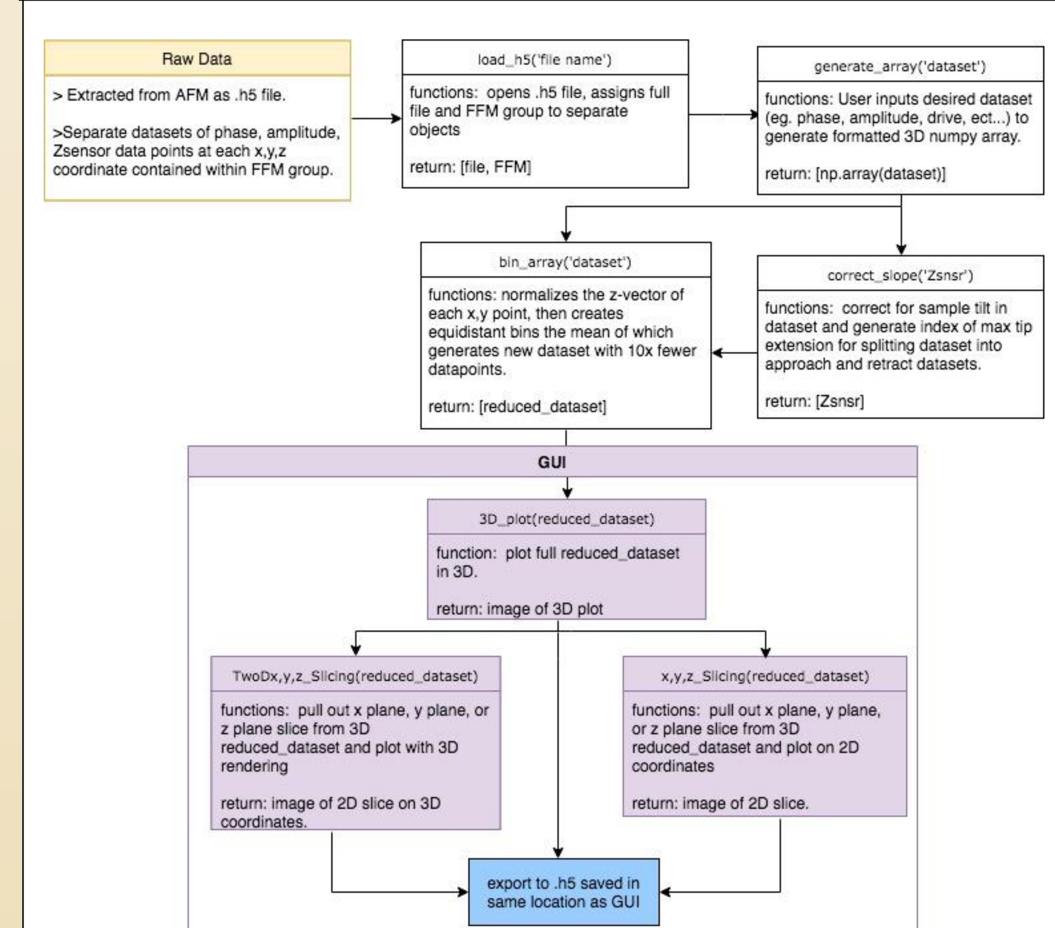
3ddatadiver has four main functions:

1) **Data Processing** to correct sample slope and generate baseline.
2) **Visualization** of the approach (Fig. 4(a)) or retract dataset. **3ddatadiver** is flexible enough to handle datasets of different sizes.
3) **Slicing** the dataset in x, y, or z plane. Users can select **any layer of interest** to see plots in **2D and 3D** cartesian coordinate systems. For the z-slicing the phase shift (or amplitude, etc) of the a selected point is displayed along with the corresponding coordinates.
4) **Exporting** of .h5 files generated in any of the aforementioned steps. This allows users to perform their own manipulations on cleaned data and/or sliced data.

GUI Design



Workflow



Our code takes advantage of a number of packages available online such as matplotlib, itertools, tkinter, h5py. In addition, we built custom functions to clean data, visualize and export data to build our GUI.

Limitations

The GUI runs slow due to large datasets and code reliance on for loops. This could be ameliorate by changing where in the code the data is processed and streamlining the code. Also, the 3D visualization is pixelated due to the limitations of matplotlib. It would be interesting to try a more sophisticated 3D rendering package, such as MayAVI.

References

[1] https://www.anasysinstruments.com/technology/nanoir-technology/
[2] Atomic force microscopy, https://en.wikipedia.org/wiki/Atomic_force_microscopy#Overview
[3] Nakouzi, Elias. 2018. PNNL

DIRECT
Data Intensive Research Enabling Clean Technologies

Acknowledgements

Special thanks to Professor Dave Beck, Elias Nakouzi, and Jim De Yoreo for their guidance.

Pacific Northwest NATIONAL LABORATORY

MATERIALS SCIENCE & ENGINEERING