

Defect identification and statistics toolbox: Automated defect analysis for scanning probe microscopy images

Alana Gudinas

University of New Hampshire, Department of Physics and Astronomy, 9 Library Way,
Durham, NH 03824, USA

Jason Moscatello

University of New Hampshire, Department of Physics and Astronomy, 9 Library Way,
Durham, NH 03824, USA³‡

Shawna M. Hollen

University of New Hampshire, Department of Physics and Astronomy, 9 Library Way,
Durham, NH 03824, USA

Abstract. Identifying and classifying defects in scanning probe microscopy (SPM) images is an important task that is tedious to perform by hand. In this paper we present the defect identification and statistics toolbox (DIST), an image processing toolbox for identifying and analyzing atomic defects in SPM images. DIST combines automation with user input to accurately and efficiently identify defects and automatically compute critical statistics. We describe using DIST for interactive image processing, generating contour plots for isolating extrema from an image background, and processes for identifying defects.

‡ Present address: Textron Systems, Wilmington, MA, USA

1. Overview

1.1. Introduction

Scanning probe microscopy (SPM) is a powerful, mature technique that enables real-space, atomic-resolution imaging of surfaces. An important role commonly played by SPM is imaging defects in order to study their role as chemically active atomic sites [1, 2, 3, 4, 5], in the assembly or growth of new crystalline layers [6, 7, 8, 9, 10, 11], and in the magnetic, electronic, and optoelectronic properties of materials [12, 13, 14]. Sometimes studies of single or small numbers of defects suffice, but when defect statistics are desirable, researchers have limited options for image analysis that can reliably and controllably generate statistics of defects in SPM images. While existing programs such as WSxM [15] and SPIW [16] provide sophisticated processing tools for SPM images, defect analysis is not automated and is time-consuming to perform by hand. To fill this need, we created a MATLAB-based image analysis toolbox that automates identification and statistical analysis of defects in SPM images [17]. The Defect Identification and Statistics Toolbox (DIST) identifies defects in SPM images with user-guided automation and outputs defect locations, counts, and other statistics, as shown in Figure 1. The defects of interest are typically atomic vacancies, substitutions, or adatoms. As long as the shape and size of the defects are fairly similar, DIST can be used to identify them and compile their statistics.

Most of the processing and analysis in DIST is performed automatically, with opportunities for user input at different stages of defect identification. Collection of defect statistics, including line profiles of each defect, are completely automated, greatly reducing user time spent on data analysis. The combination of automation and user input makes for fast, efficient, and accurate defect identification and analysis. In this paper, we present 1) an overview of DIST; 2) a step-by-step explanation of how the program works, including inputs, outputs, and user controls; and 3) a description of limitations and possible expansions.

1.2. Program pathway

Figure 2 outlines the main processes in DIST and the relevant subprocesses. After the image is first processed through the GUI (Figure 3)), the user chooses to either identify bright or dark defects with respect to the background of the processed image displayed in the GUI. DIST uses MATLAB's contour-generating function to identify defect candidates in the image. Since this procedure will recognize any features distinct from the background as defect candidates, user input is needed to narrow the selection to defects of interest. To do this, the user can choose 1) shape-matching algorithms (Figure 4), or 2) size and apparent height filtering. For both methods, the user selects a representative defect to be used as a template to which other defect candidates will be compared. The program then identifies defects that match the template within defined tolerances.

Since the user is the expert on differentiating between the defects of interest and other defects or artifacts in the image, it is crucial that users are able to choose the level of processing best suited for an image, customize analysis on regions in the image, and add or delete defects

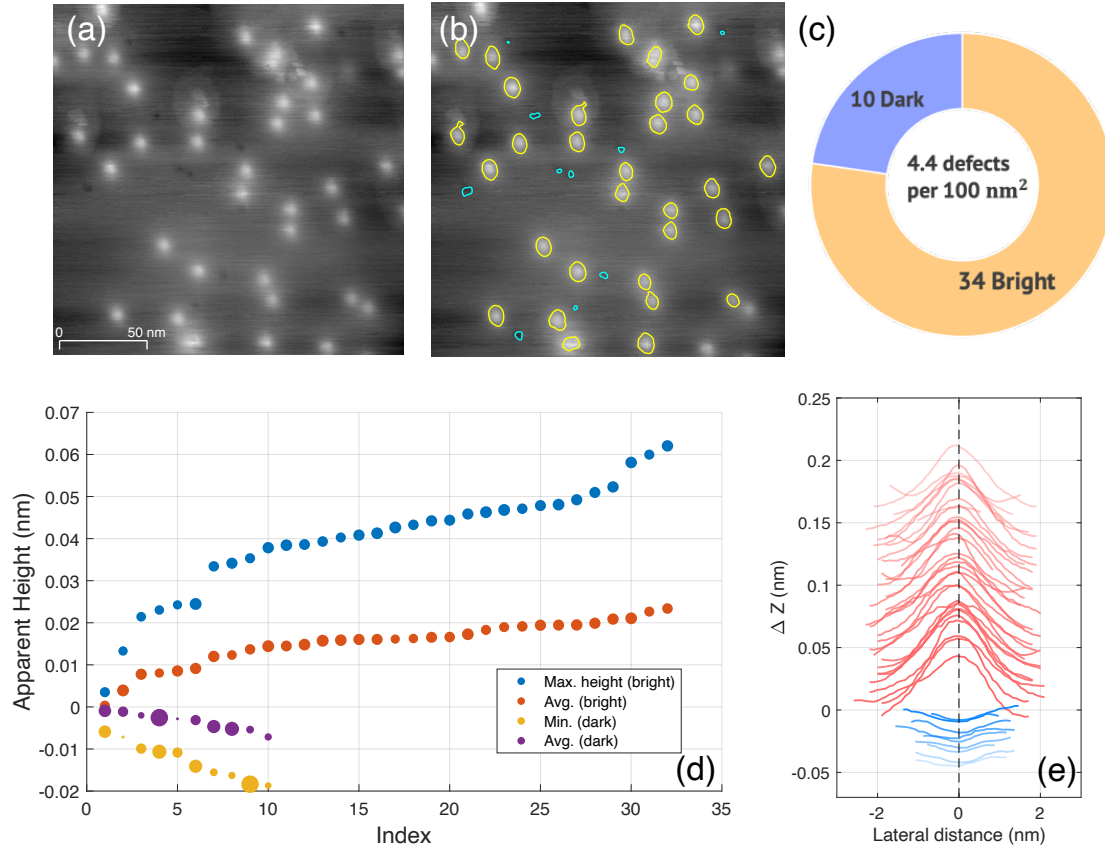


Figure 1. Automated defect identification and statistics a) Line-flattened STM image of MoTe₂. b) Line-flattened image with positively identified defects outlined with contours. The yellow contours identify bright defects, and the blue identify dark defects. c) Defect count and defect density. d) Apparent height data, indexed by increasing (decreasing) brightness for bright (dark) defects. e) Line profiles generated by DIST across the major axis of each defect contour. Line profiles are each offset by 5×10^{-3} nm for clarity.

to correct the automated identification. The options for user correction are: 1) Applying a secondary filter (when contour filtering is the specified defect identification process); 2) Adding or deleting contours to more correctly represent defects; 3) Selecting regions in the image that require new or further processing to accurately isolate defects. If, after the defect identification process is finished, there are defects that have not been identified, this function allows users to re-process specific regions of the image to identify remaining defects.

After a set of defects are identified, their statistics are generated. The user specifies the direction of the spatial cross-section of the defects for generating line profiles, across either the major or minor axis. The primary statistics that DIST returns are: X and Y coordinates of all identified defects, line profiles, apparent height, and area, as shown in Figure 1.

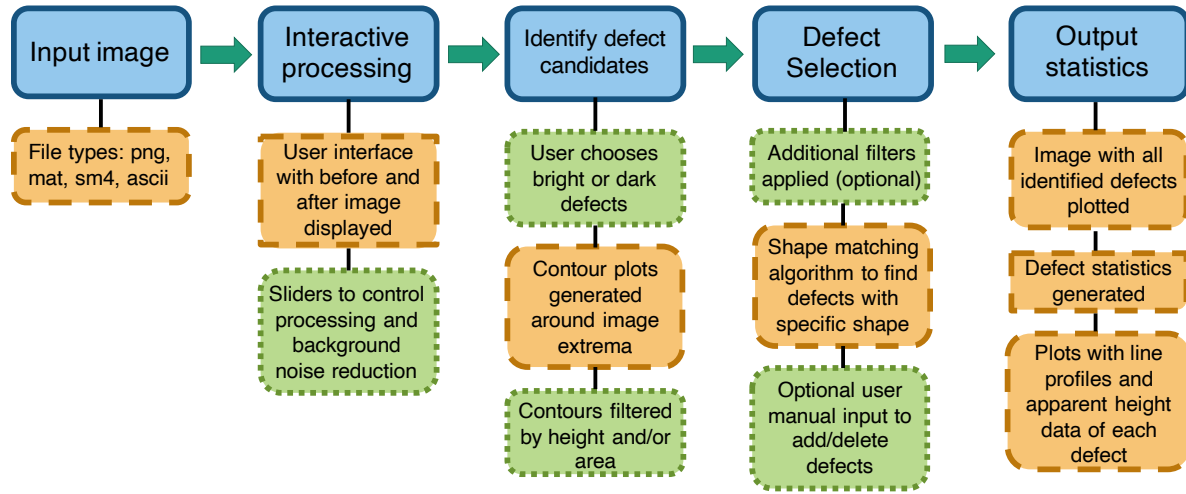


Figure 2. Flow diagram of program pathway Blue/solid line boxes indicate the main steps of DIST, and the subcategories describe the processes in each section. Orange/dashed line boxes indicate automatic processes, and green/dotted line indicate manual options.

2. Program description

2.1. Input Requirements

DIST accepts the following file types for analysis: .sm4, .mat, .png, and .asc. Any other file type can also be used if it is first converted into a .mat file. If the input file is .sm4, the image is converted into a .mat structure with attributes such as image width included. If the file is not in .sm4 form, the width of the image in nanometers is additionally required as an input.

2.2. Image processing

Noise and nonuniform background topography make automated identification of defects in an image difficult. These challenges also vary from image to image, so image processing procedures need to be adjustable. DIST provides a GUI with interactive slider controls for the image processing parameters (Figure 3). Upon opening the program, the raw image is flattened by subtracting a line-by-line linear fit from the image data. Then four processing controls are applied according to the slider positions (in order): 1) "strel" object radius, 2) gaussian filter, 3) lower bound, and 4) upper bound. The user begins with the line-flattened image and adjusts the sliders to optimize the image for defect identification. Below are descriptions of each processing parameter.

- (i) Strel: MATLAB's morphological structuring element. Strel generates disks with a user-specified radius in pixels. MATLAB's "imopen" function then opens the image with the structuring element and the user controls the radius of the strel object using a slider, while the processed image updates in real time. Each time the value of the strel radius is changed, the image created from the strel object is subtracted from the original line-flattened image. This function is especially useful for images with nonuniform background topography.

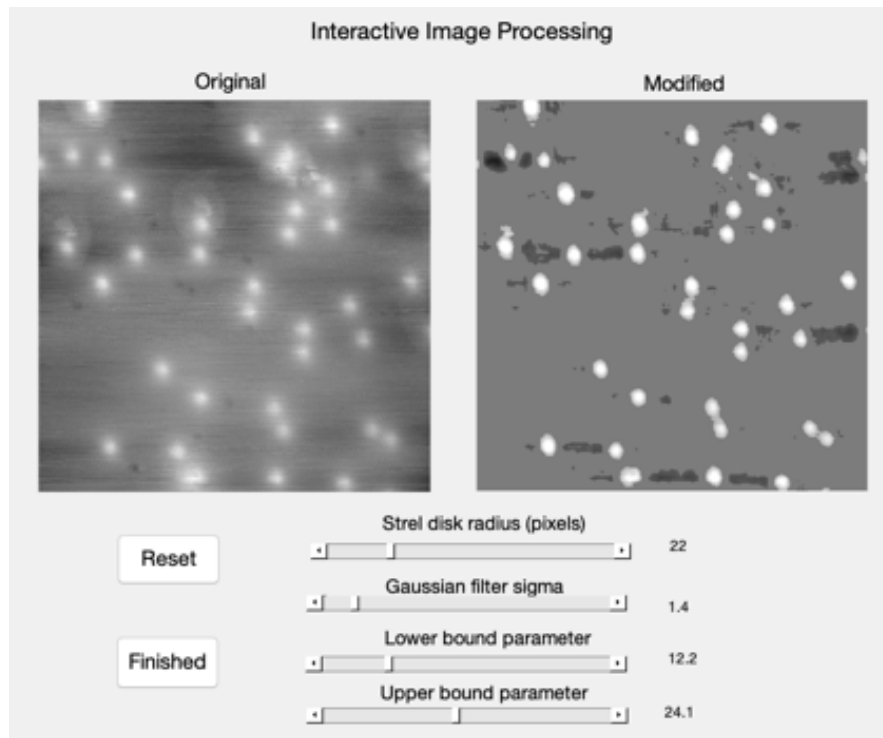


Figure 3. A snapshot of the MATLAB graphical user interface (GUI) a) Line-flattened image. b) Result of interactive processing sliders.

- (ii) Gaussian filter: smooths the image using a Gaussian filter of a specified width (standard deviation). The user controls the width using the slider. A larger (smaller) value results in a correspondingly large (small) Gaussian spatial averaging. This parameter is useful especially for images with grainy or high-frequency noise.
- (iii) Lower/upper bound: parameters to determine the range of pixel values in the image that define the background. A slightly simpler method of defining an image background is via "flooding," which sets a threshold brightness value that a feature must exceed in order to be considered part of the foreground. Problems with flooding arise when the brightness of image features are nonuniform, and applying a single threshold value destroys valuable information in the image. The earlier processing steps partially address this. Additionally, we implement two brightness parameters, rather than one, to set a lower and upper bound on the image background. Both parameters can be adjusted independently in order to customize the background for each image. If both values are set to zero, every pixel value in the image is considered part of the foreground. When the lower and upper bounds are increased, any pixel values that are brighter than the lower bound parameter and darker than the upper bound parameter are set to the mean pixel value of the whole image. What remains are the brightest and darkest extrema in the image, while the rest of the image is a uniform brightness (Figure 4(a)).

2.3. Defect identification

2.3.1. Contour Filters The defect identification process begins when the user chooses either bright or dark extrema. MATLAB's built-in contour generating function then creates a topographical-like map on the processed image, where contours are generated around all features that are brighter (or darker) than the image mean pixel value. The brightness gradient in the image is divided into 20 values, or brightness levels, so that each contour line corresponds to a specific brightness value in the image. We have found that 20 brightness levels is a good balance between having too many or too few contours in the majority of images. Having too many levels introduces an unnecessary amount of detail for defining defect boundaries and increases the computational time needed to analyze the contours, while having too few will not accurately capture defect boundaries.

Since extrema in the processed image are isolated from the background following the interactive user-processing, potential defects will be enclosed by rings of contours. Before precisely identifying defects, the contours are filtered to limit the pool. The user can filter contours based on their attributes: contour area, brightness level, and minimum vertex count. The contour area value is computed from the difference in area between the template defect selected by the user and all other contours. The vertex count refers to the number of points that comprise each contour plot. This crude filtering process reduces the pool of contours used for defect identification, which makes further computation faster. The results of contour filtering are plotted in Figure 4(a) and Figure 1(b). For images with defects that do not vary much by brightness or size, most of the defect candidates can be isolated through crude filtering alone. After filtering, multiple contours may encircle one potential defect. In this case the outer contour(s) typically capture the outermost edges of the defect, where there is less of a distinction between the background and the defect, while the inner contours capture the actual shape of the defect where the brightness is most extreme as compared to the background. Thus, in the case of multiple contours per defect after filtering, the inner contour is always chosen.

2.3.2. Shape Matching After the filtering process is completed, the remaining contours constitute the pool from which defects will be identified. At this point, the defects are identified through either a second round of filtering identical to the process outlined in 2.3.1 or a shape-matching algorithm. To conduct shape-matching, DIST uses an algorithm written by Oliver Van Kaick et al. to perform contour correspondence using ant colony optimization (ACO)[18]. First, each defect candidate is enclosed by a cluster of contours (Figure 5(a)). DIST then prompts the user to draw a rectangle around a defect with a shape of interest. The set of contours enclosed in the rectangle is displayed in a second window, where the user can then select the contour that most closely resembles the shape of the defect. The selected contour (seen in Figure 4(b)) serves as the template defect to which all other contours will be compared.

The ACO shape-matching algorithm [18] is a type of quadratic assignment problem (QAP) that compares two contours. The ACO approach emulates how ants use pheromones

to let the rest of the colony know where food is located. The algorithm determines the contour correspondence by allowing agents (“ants”) to traverse possible paths between vertices defining two contours and deposit virtual pheromones according to the degree the contours match (see Figure 4(c) for a visual example). The cost of matching between two contours is computed with the QAP objective function and reflects how similar two contours are in shape. The best cost is used in DIST to sort contours based on how close they are in shape to the user-selected reference contour. The ACO algorithm is unique from other correspondence optimization algorithms because it incorporates both order-preservation (maintaining vertex order), and proximity information, which ensures that a pair of vertices that outline a feature on one contour is mapped to the same feature pair on another contour. Because ACO searches for optimum paths between two contours, it is orientation and size independent.

In DIST, shape matching is applied first within each cluster of contours so that the contour that best matches (lowest cost) the shape of the template is selected and the remainder are discarded. This process ensures that only one contour per defect is saved. Figure 5 shows how all the contours in a cluster compare to the shape matching criteria.

Once the best-matching shape from each contour cluster is determined, the user may filter the identified defect candidates for shape using the best cost value. If the user is only interested in shapes that very closely resemble the target shape, they may identify a cut-off point in a histogram of best cost values, after which point defects with a higher cost value will be removed from the pool. The results of both the shape-matching process and filtering by the best cost parameter are shown in Figure 4. Figure 5(a) is an example of a perfect shape match, panel (b) is an example of a contour with an acceptable shape, and panel (c) is an example of a contour that falls beyond the best cost parameter threshold and is rejected.

2.4. Outputs

DIST automatically computes a number of statistics after all the defects in the image have been identified. Through MATLAB’s “regionprops” function, the centroid coordinates, contour areas, eccentricity value, lengths of major and minor axes, and orientation of each defect are computed. The major and minor axes and orientation data are used to generate line profiles across each defect, which are automatically plotted as in Figure 1(e). The line-flattened image with the identified contours is also displayed automatically.

Additionally, DIST calculates the maximum and mean apparent height across each line profile (relative to the background). The apparent height data is automatically plotted from least to greatest to demonstrate the defect brightness associated with each layer of the sample (see Figure 1(d)). Perhaps the most valuable of all the outputs are the X and Y coordinates data of every defect, organized into their own matrices. With access to the contour coordinates, users can generate their own statistics as well. The combined output statistics give users increased flexibility in manipulating the defect data after identification. See Table 1 for a list of all the statistics automatically compiled. The statistics are output to the MATLAB workspace, where the user can choose to save and display the data of interest.

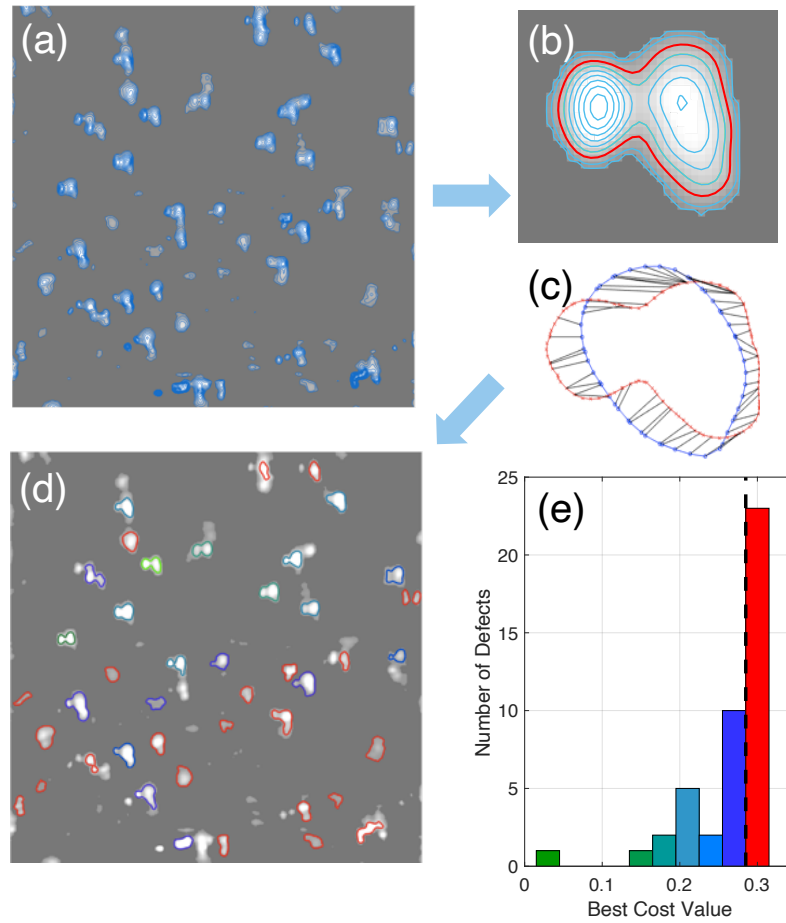


Figure 4. Shape matching a) Processed STM images with filtered contour plots. Contours with an area below a certain threshold were excluded. b) Close-up example of a contour cluster. The blue curves indicate the contour cluster, and the red curve indicates the selected template. c) Example of contour correspondence during shape-matching process. The red plot indicates a template contour, and the blue plot is the shape being compared. The black lines connecting the matching vertices of the two contours are the results of the contour correspondence algorithm. d) Image with defects identified after the shape-matching process. Plots are color-coded according to the histogram in (e). e) Histogram plotting the best cost value for each of the contours displayed on the image in (d). Green indicates the lowest cost value, i.e., the closest match, and red indicates contours that were excluded after filtering the results by best cost values. The black dashed line indicates the threshold best cost value.

Table 1. Defect statistics compiled by DIST

Computed statistics	regionprops statistics
Apparent height	Area
Line profile	Eccentricity
X/Y coordinates	Major/minor axis
Defect contours	Centroid
	Orientation

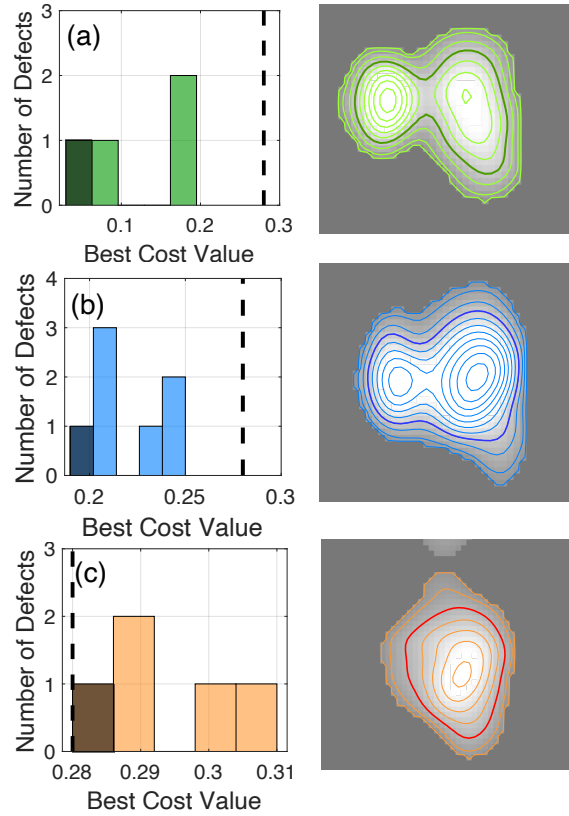


Figure 5. Shape filtering (a) Results of shape-matching for a perfect match (i.e., comparing the template shape with itself). (b) Example of a shape that is within the best cost parameter threshold, indicated by the black dashed line. (c) Example of a shape that is beyond the best cost parameter threshold and rejected. (The best matching contour in the cluster has a best cost value above the threshold.) In each panel, the dark histogram bar corresponds to the dark contour in the image.

3. Flexibility and development

3.1. Manual control

For images with high levels of background noise that make defects hard to distinguish, DIST might misidentify defects or fail to isolate defects. Therefore, there are points in the program at which the user can provide input to make defect identification more accurate. If, after both shape-matching and contour-filtering, there are unidentified defects in the image, the user may opt to use a rectangle to select a region that needs further analysis. By choosing a small region of the image to be analyzed, the contour plots generated around the potential defect are finer in resolution. The user may then choose shape-matching or filtering to compare the contours generated in the small region, or simply manually select a contour that encircles a defect.

Additionally, there may be defects that are incorrectly identified. Before defect statistics are computed, the user may delete contours surrounding incorrectly identified defects. The last step before defect analysis allows the user to "quick add" any contours around defect

candidates that are not yet included in the set. This is a faster process than re-processing a region of the image, because the user can simply pick contours by eye that were missed in the automated identification.

3.2. Examples of use

Here we present several examples of use of the toolbox, including images with atomic resolution and selected images from the literature. We began analysis using .png files and employed various processing techniques available in DIST to identify defects in the images. The defect identification results are shown in Figure 6. Below is a description of the processing for each image as well as the time spent using DIST.

Figure 6(a) is an atomically-resolved image of black phosphorus 18 nm wide. There are four bright defects present in the image, three of which are not cut off by the image boundary. To process this image for defect identification, we applied a large Gaussian filter using the image processing GUI (Section 2.2) to blur the atomic resolution. Blurring the fine image details resulted in a more uniform brightness in the defect regions, so that generated contours encapsulate the entire defect. After the Gaussian filter, the lower bound parameter (Section 2.2) was adjusted to the maximum value to leave only the bright extrema in the image. The upper bound parameter was then slowly adjusted to remove the lighter background surrounding the defects, leaving only the bright defect areas. After image processing, the defects were selected from the contours using area filtering (Section 2.3.1). Since there are no other major artifacts in the image, all three defects were identified by contours after the first round of filtering, and no other defects needed to be manually added or deleted. Processing this image took approximately three minutes.

Figure 6(b) is an image of epitaxial graphene from [19] with atomic resolution. In the image processing GUI, the lower bound parameter was adjusted to the maximum value since the defects are brighter than the background. As with the image in Figure 6(a), the upper bound parameter was slowly adjusted until the defects were isolated from the background. The lattice structure visible in the image was reduced by adjusting the upper bound, since the structure is brighter than the background but not as bright as the defects. After image processing, the defects were identified with area filtering. The first round of filtering captured all of the defect contours as they are fairly uniform in size. No manual adding or deleting of contours was necessary. Processing this image took approximately three minutes.

Figure 6(c) is an image of MoS₂ with bright defects in which the atomic lattice is clearly resolved. Here, the maximum Gaussian filter was applied to blur the atomic lattice. The contours were isolated using area filtering (Section 2.3.1). The generated contours captured most of the potential bright defects in the image as well as some non-defect artifacts. In areas with large bright spots, the contours did not isolate the defects. After filtering, eight contours missed by the area filter were manually added by re-analyzing specific regions of the image, and three non-defect contours were manually deleted (shown in Figure 6(c)). Including the manual steps, this image took approximately 10 minutes to process.

Figure 6(d) is an image of Sb₂Te₃(0001) from [21]. Here, the different defects have

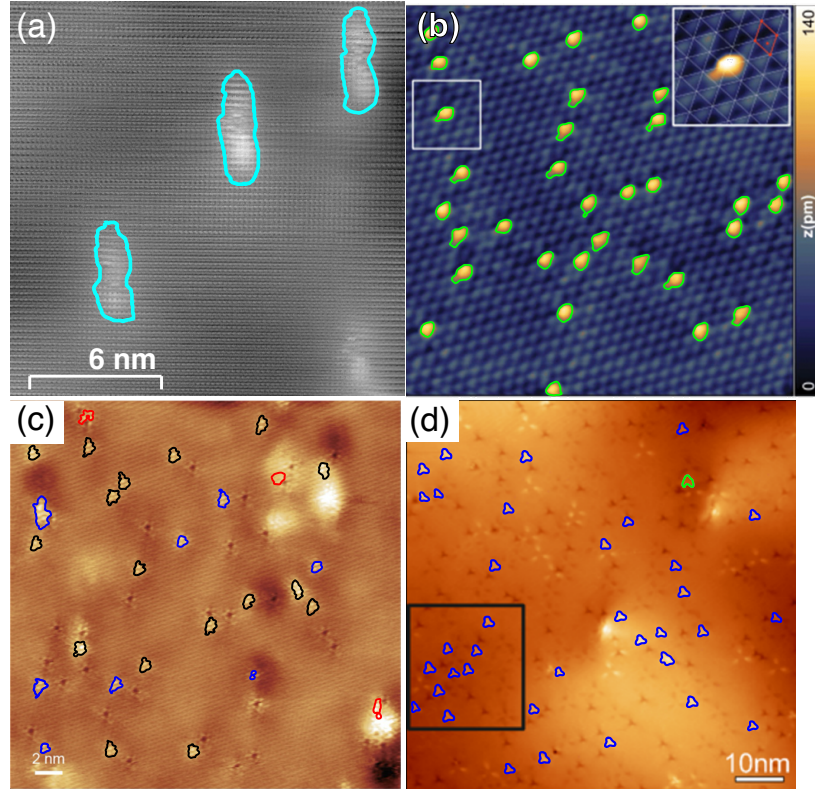


Figure 6. Examples of defect identification using DIST. a) Atomically resolved image of defects in black phosphorus. b) Image of epitaxial graphene with mostly uniform defects from [19]. c) Image of MoS_2 with defects and defect clusters from [20]. Red indicates contours that were manually deleted, and blue indicates contours that were manually added. d) Image of $\text{Sb}_2\text{Te}_3(0001)$ from [21] with bright, triangle-shaped defects identified. Green indicates a manually deleted contour.

distinct shapes, making this image a good candidate for shape-matching (Section 2.3.2). After isolating the bright defects by a similar process as in Figure 6(a) and (b), we used shape matching to isolate the bright triangle-shaped defects. A reference triangular contour was chosen for comparison. The shape-matching process took 10 minutes to complete, after which the results were filtered based on best cost values, as in Figure 4 (Section 2.3.2). After the best cost filter, there remained one contour that was manually deleted (Figure 6(d)). The total time elapsed processing this image was approximately 18 minutes. This example shows that while filtering contours is typically faster for defect identification, there are cases where shape-matching and filtering by shape similarity is useful for isolating defects with a distinct shape.

3.3. Limitations and future development

DIST is limited in part by the quality of the STM images. For images where defects are difficult to distinguish even by eye, it is likely that the image processing will fail to isolate extrema from the background without also losing defect information. Additionally, in such

images contour plots may not accurately encircle defects. DIST also relies on a relatively uniform background. Images with defects on separate crystalline terraces may need special treatment, such as analyzing terraces separately or first applying a derivative filter, in order for DIST to work well.

Due to the flexibility of DIST, there are many features that could be added to the program in the future. A function that can track how the shape and location of a defect changes over the course of multiple images could be implemented, providing users with data on defect dynamics. Implementing a smooth way to do batch processing and analysis on many similar images would also improve the program and lower analysis time for users.

Any future changes to DIST will be updated on GitHub [17] and the Hollen Lab website. We plan to upload the complete toolbox to MathWorks.

Acknowledgments

We are grateful for input on the program and example images collected in collaboration with Jay Gupta and Sara Mueller of The Ohio State University Department of Physics and Center for Emergent Materials. This material is based upon work supported by the National Science Foundation under Grant No. DMR-1709029.

The data that support the findings of this study are available from the corresponding author upon reasonable request.

- [1] Seung Mi Lee, Young Hee Lee, Yong Gyoo Hwang, J. R. Hahn, and H. Kang. Defect-induced oxidation of graphite. *Physical Review Letters*, 82(1):217–220, January 1999.
- [2] Werner Weiss and Wolfgang Ranke. Surface chemistry and catalysis on well-defined epitaxial iron-oxide layers. *Progress in Surface Science*, 70(1-3):1–151, March 2002.
- [3] Ulrike Diebold. The surface science of titanium dioxide. *Surface Science Reports*, 48(5-8):53–229, January 2003.
- [4] A. C. Papageorgiou, N. S. Beglitis, C. L. Pang, G. Teobaldi, G. Cabailh, Q. Chen, A. J. Fisher, W. A. Hofer, and G. Thornton. Electron traps and their effect on the surface chemistry of TiO₂(110). *Proceedings of the National Academy of Sciences*, 107(6):2391–2396, January 2010.
- [5] J.V. Lauritsen and F. Besenbacher. Atom-resolved scanning tunneling microscopy investigations of molecular adsorption on MoS₂ and CoMoS hydrodesulfurization catalysts. *Journal of Catalysis*, 328:49–58, August 2015.
- [6] J. Libuda, F. Winkelmann, M. Bäumer, H.-J. Freund, Th. Bertrams, H. Neddermeyer, and K. Müller. Structure and defects of an ordered alumina film on NiAl(110). *Surface Science*, 318(1-2):61–73, October 1994.
- [7] M. Ritter, W. Ranke, and W. Weiss. Growth and structure of ultrathin FeO films on pt(111) studied by STM and LEED. *Physical Review B*, 57(12):7240–7251, March 1998.
- [8] S. Fölsch, A. Helms, S. Zöphel, J. Repp, G. Meyer, and K. H. Rieder. Self-organized patterning of an insulator-on-metal system by surface faceting and selective growth: NaCl/cu(211). *Physical Review Letters*, 84(1):123–126, January 2000.
- [9] E. Wahlström, N. Lopez, R. Schaub, P. Thosttrup, A. Rønnau, C. Africh, E. Lægsgaard, J. K. Nørskov, and F. Besenbacher. Bonding of gold nanoclusters to oxygen vacancies on RutileTiO₂(110). *Physical Review Letters*, 90(2), January 2003.
- [10] Niklas Nilius. Properties of oxide thin films and their adsorption behavior studied by scanning tunneling microscopy and conductance spectroscopy. *Surface Science Reports*, 64(12):595–659, December 2009.
- [11] M. M. Ugeda, I. Brihuega, F. Guinea, and J. M. Gómez-Rodríguez. Missing atom as a source of carbon magnetism. *Physical Review Letters*, 104(9), March 2010.

- [12] Oleg V. Yazyev and Steven G. Louie. Topological defects in graphene: Dislocations and grain boundaries. *Physical Review B*, 81(19), May 2010.
- [13] H. Gonzalez-Herrero, J. M. Gomez-Rodriguez, P. Mallet, M. Moaied, J. J. Palacios, C. Salgado, M. M. Ugeda, J.-Y. Veuillen, F. Yndurain, and I. Brihuega. Atomic-scale control of graphene magnetism by using hydrogen atoms. *Science*, 352(6284):437–441, April 2016.
- [14] Sara Barja, Sivan Refaely-Abramson, Bruno Schuler, Diana Y. Qiu, Artem Pulkin, Sebastian Wickenburg, Hyejin Ryu, Miguel M. Ugeda, Christoph Kastl, Christopher Chen, Choongyu Hwang, Adam Schwartzberg, Shaul Aloni, Sung-Kwan Mo, D. Frank Ogletree, Michael F. Crommie, Oleg V. Yazyev, Steven G. Louie, Jeffrey B. Neaton, and Alexander Weber-Bargioni. Identifying substitutional oxygen as a prolific point defect in monolayer transition metal dichalcogenides. *Nature Communications*, 10(1), July 2019.
- [15] I. Horcas, R. Fernández, J. M. Gómez-Rodríguez, J. Colchero, J. Gómez-Herrero, and A. M. Baro. WSXM: A software for scanning probe microscopy and a tool for nanotechnology. *Review of Scientific Instruments*, 78(1):013705, January 2007.
- [16] Julian Stirling, Richard A. J. Woolley, and Philip Moriarty. Scanning probe image wizard: A toolbox for automated scanning probe microscopy data analysis. *Review of Scientific Instruments*, 84(11):113701, November 2013.
- [17] DIST Access, <https://github.com/alanagudinas/Hollen-Lab>.
- [18] Oliver van Kaick, Ghassan Hamarneh, Hao Zhang, and Paul Wighton. Contour correspondence via ant colony optimization. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*. IEEE, October 2007.
- [19] Virginia Carnevali, Laerte L. Patera, Gianluca Prandini, Matteo Jugovac, Silvio Modesti, Giovanni Comelli, Maria Peressi, and Cristina Africh. Doping of epitaxial graphene by direct incorporation of nickel adatoms. *Nanoscale*, 11:10358–10364, 2019.
- [20] Xiaolong Liu, Itamar Balla, Hadallia Bergeron, and Mark C. Hersam. Point defects and grain boundaries in rotationally commensurate mos2 on epitaxial graphene. *The Journal of Physical Chemistry C*, 120(37):20798–20805, 2016.
- [21] C. Pauly, C. Saunus, M. Liebmann, and M. Morgenstern. Spatially resolved landau level spectroscopy of the topological dirac cone of bulk-type $\text{sb}_2\text{te}_3(0001)$: Potential fluctuations and quasiparticle lifetime. *Phys. Rev. B*, 92:085140, Aug 2015.