doi:10.1017/S1431927619001612

Advancements to the Digital Representation Environment for Analysis of Materials in 3-Dimensions—DREAM.3D

M. Jackson[1*], M. Groeber[2,1], S. Donegan[3] and D. Dimiduk[1]

[1.] BlueQuartz Software, LLC, Springboro, OH, USA.
[2.] Department of Integrated Systems Engineering, The Ohio State University, Columbus, OH, USA.
[3,] Air Force Research Laboratory, Materials and Manufacturing Directorate, AFRL/RXCM Bldg 655, Wright-Patterson AFB, OH, USA.
* Corresponding author: mike.jackson@bluequartz.net

More than 15 years ago, researchers at the Air Force Research Laboratory (AFRL), together with sponsors from DARPA and AFOSR, explored methods for directly including 3-dimensional (3D) microstructure information into engineering design systems [1]. In time, advancements led to BlueQuartz Software developing DREAM.3D (D3D) for public use in 2012 [2]. Two early use modes of D3D centered on processing experimental 3d-EBSD data to perform virtual reconstructions, and generation of "synthetic," statistically-equivalent, virtual materials using quantitative microstructure information for predictive simulations. To foster the broadest use of these capabilities, D3D was introduced as an open-source project, enabling an international community of users and contributors.

The core capabilities of DREAM.3D continue to expand and become more robust. For example, the now fully integrated StatsGenerator filter allows users to methodically create statistics that will lead to a microstructure of their design. The user can also use statistics that have been calculated from experimental data. Other recent filters ease the coupling of materials models from DREAM.3D to response simulation codes, as well as for importing simulation data from those codes. Since 2014 other programs/projects stimulated advances to the Filter–Pipeline analysis framework, and these extended DREAM.3D into several lesser-known application domains. DREAM.3D has been adapted as a web service for performing analyses of large 3D data sets without moving the data itself [3]. Further, the methods for acquiring large-scale 3D materials data are inefficient and expensive. By adapting DREAM.3D for real-time image analysis, Godaliyadda, et al., demonstrated a sophisticated targeted sampling method for efficient image acquisition [4]. Also, Groeber, et al., demonstrated using DREAM.3D for data fusion and analysis for understanding processing parameters and resultant metal additive manufacturing builds [5].

Perhaps of greater interest for microscopy are the expanding DREAM.3D capabilities for grey scale image analysis and image montaging. Recently, BlueQuartz collaborated with Kitware, Inc and AFRL to integrate over 100 filters from the Insight Segmentation and Registration Toolkit (ITK) that allow the processing of images (data on a regular grid). Analyses include traditional image processing, such as intensity transformations and kernel convolutions, along with more advancement processing, including denoising, multi-scale object identification, and segmentation. Current collaborative developments are integrating a new ITK module called "ITKImageMontaging" that allows registration, background leveling, and stitching of wide area image mosaics. Building on the prior work of Preibisch, et al. [6], and Lykar, et al. [7], the effort intends to construct a general and robust toolset for 3D montage datasets. An example use-case of this toolset is show in Figure 1.

Peering into the future reveals other nascent but major advances to DREAM.3D. For example, the DREAM.3D are currently leveraging the Pybind11 project to automatically create Python modules that

https://doi.org/10.1017/S1431927619001612 Published online by Cambridge University Press

allow DREAM.3D libraries to be accessed from Python codes. These tools allow for building integrated pipelines into a larger workflow using Python. Another emerging project has implemented a RESTful API, allowing D3D to be incorporated into enterprise-wide digital data solutions as a remote resource. Finally, the coming year will see the image montaging tools expanded to include montaging of EBSD data maps from large area.

There are many organizational aspects of the DREAM.3D project itself that need to be considered in order to keep the infrastructure up to date, including: where the code is stored, how the code is updated and documented, and when/how binaries are released. For collaborative development, DREAM.3D uses GitHub to store source code, adopting GitHub's standard "pull request" (PR) model where a contributor can request DREAM.3D developers to review their code and ultimately include the code into the publicly released version of DREAM.3D. The licensing model that BlueQuartz adopted for DREAM.3D permits free use the tools behind "closed doors" of commercial companies, even permitting inclusion proprietary algorithms without the need to publish the codes. BlueQuartz Software maintains Google Groups for both DREAM.3D users and developers in addition to an array of computer systems used for nightly build, testing and packaging of the DREAM.3D application.

In summary, DREAM.3D has evolved into a widely used open innovative platform for processing and analysis of spatiotemporal materials information. The project would benefit from still greater community wide contributions [8].

References:

[1] DM Dimiduk et al., in "Materials Processing and Design: Modeling, Simulation, and Applications", (Springer-Verlag, NY) p. 1705.

[2] A Groeber and M Jackson, Integrating Materials and Manufacturing Innovation **3** (2014), p. 1.

[3] T Pollock et al., Private Communication (2018).

[4] GM Dilshan Godaliyadda et al., IEEE Transactions on Comp. Imaging XIV **4** (2018), p. 1.

[5] M Groeber et al., IOP Conference Series: Materials Science and Engineering **219 (1)**, p. 012002.

[6] Preibisch et al., Bioinformatics **25** (2009), p. 1463.

[7] Lykar et al., Jol of Microscopy **197** (200), p. 285.

[8] The authors acknowledge funding under USAF prime contract FA8650-15-D-5231.

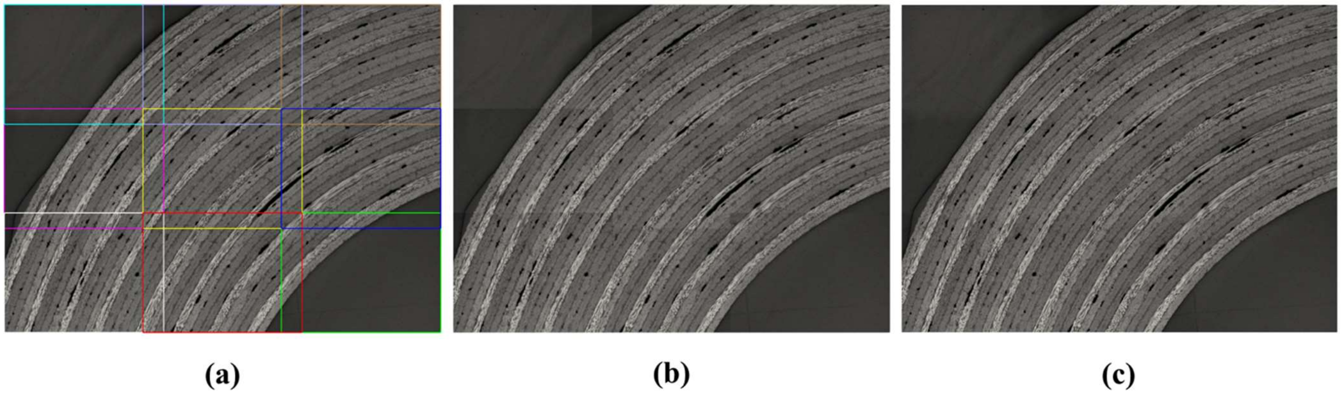

Figure 1. The left panel (a) shows the outlines of the individual images that make up the mosaic, the center panel (b) shows a stitched mosaic from the individual tiles with no attempt to fine adjust the positions of the tiles or blend the tiles in the overlap areas. The right panel (c) shows a complete mosaic with subpixel adjustment to the tile positions and an appropriate blending algorithm applied.

https://doi.org/10.1017/S1431927619001612 Published online by Cambridge University Press