Structure Recognition from High Resolution Images of Ceramic Composites

Daniela Ushizima[1,4], Talita Perciano[1], Harinarayan Krishnan[1], Burlen Loring[1], Hrishikesh Bale[2], Dilworth Parkinson[3] and James Sethian[1]
[1]Computational Research Division, [2]Material Science Division, [3]Advanced Light Source Division,
Lawrence Berkeley National Laboratory, Berkeley, CA and
[4]Berkeley Institute for Data Sciences, University of California Berkeley

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

This work was partially supported by the Office of Energy Research, U.S. Department of Energy, under Contract Number DE-AC02-05CH11231.

Structure Recognition from High Resolution Images of Ceramic Composites

January 5, 2015

Daniela Ushizima[1,4], Talita Perciano[1], Harinarayan Krishnan[1], Burlen Loring[1], Hrishikesh Bale[2], Dilworth Parkinson[3] and James Sethian[1]
[1]Computational Research Division, [2]Material Science Division, [3]Advanced Light Source Division, Lawrence Berkeley National Laboratory, Berkeley, CA and
[4]Berkeley Institute for Data Sciences, University of California Berkeley

Abstract

Fibers provide exceptional strength-to-weight ratio capabilities when woven into ceramic composites, transforming them into materials with exceptional resistance to high temperature, and high strength combined with improved fracture toughness. Microcracks are inevitable when the material is under strain, which can be imaged using synchrotron X-ray computed micro-tomography (μ-CT) for assessment of material mechanical toughness variation. An important part of this analysis is to recognize fibrillar features. This paper presents algorithms for detecting and quantifying composite cracks and fiber breaks from high-resolution image stacks. First, we propose recognition algorithms to identify the different structures of the composite, including matrix cracks and fibers breaks. Second, we introduce our package F3D for fast filtering of large 3D imagery, implemented in OpenCL to take advantage of graphic cards. Results show that our algorithms automatically identify micro-damage and that the GPU-based implementation introduced here takes minutes, being 17x faster than similar tools on a typical image file.

Keywords: Material Inspection; Fiber Detection; ImageJ/Fiji plug-in; GPU; OpenCL.

1 Introduction

The ability to design tough ceramic matrix composites has emerged in the past decade, resulting in composites resistant to chemically corrosive conditions, extreme temperatures and brittleness. Assessing the scale, geometry and structural composition of such materials is of paramount importance to quantify strength, to monitor crack propagation and to guarantee engineering safety [1, 2]. In order to retrieve internal structure of solid objects and evaluate their properties, materials have been examined using computed microtomography (μ-CT) generated at light sources. These imaging facilities use high fluxes from synchrotron source to reconstruct three-dimensional high spatial resolution, which can be used to measure microdamage when applying different loads to composites.

Some of the major challenges in interpreting μ-CT require 3D image analysis methods capable of detecting relevant patterns that allow comparisons among samples [3]. One goal of these methods is to extract structural descriptors from 3D images that can be translated into microdamage quantifiers. For example, our automated fiber recognition module recovers the connectivity throughout the material, so one can measure the ceramic deformation under different loads.

First, in this paper we introduce a set of 3D image tools to quantify ceramic composite structure. Our software package includes algorithms to enhance images, automatically separate heterogenous composites into constituent phases, detect cracks, and identify fiber breaks at micrometer resolution.

Second, 3D image analysis at scale requires parallel-capable algorithms that can accommodate large data size and complexity, and allow real-time feedback. One approach is to use Fiji [4], a free image analysis framework for quantitative microscopy that supports processing of image stacks (3D representation), with several multi-threaded algorithms, so time-consuming operations can quickly run in parallel on multi-CPU hardware. However, while this

approach can be quite powerful, the benefits of such a threading strategy are no longer sufficient to keep up with the high quality complex images that come from photon source facilities.

To the best of our knowledge, there are no public software tools capable of segmenting tubular structures and detecting microdamage from μ-CT images, particularly when the dataset is larger than the available RAM.

In order to scale our recognition algorithms, we have developed "F3D", which implements our new software as a graphics-card aware image processing Fiji plug-in that handles 3D images greater than the amount of RAM, expediting pre-processing operations using stencil-based filters, and mathematical morphology (MM) operators. This paper describes our work on parallelized kernel-based algorithms to improve the performance of software available in the public domain.

The key contributions of this work are: (a) design of recognition algorithms for finding fibrillar/tubular structures from μ-CT, which are essential to quantify evolving microdamage; (b) parallelization of image processing codes for fast real-time processing on multi-GPUs, implemented as a hybrid of OpenCL/Java, which we named F3D; (c) performance evaluation and comparison with tools previously available in Fiji/ImageJ.

Sec. 2 describes the materials imaged into μ-CT stacks that are input to the proposed pattern recognition pipeline, described in Sec.3, which is tailored to detect individual fibers from the μ-CT datasets. Sec.4 presents implementation details about the new F3D plug-in for image processing and analysis. Sec.5 presents the performance evaluation of the proposed plug-in with current available tools, and the validation of our algorithms by applying them to μ-CT samples for detection of fiber loci, crack opening displacements and information of broken fibers with increasing applied loads. Sec. 6 and Sec. 7 provide discussions, conclusions and future works.

2 Characterizing damage in ceramic composites

Research on 3D imaging of SiC-based composites under load can reveal how microscopic damage influences strength and toughness of materials [1]. In order to maximize material strength, continuous fiber bundles are woven into the ceramic, and oriented along primary load paths. The 3D architecture of woven composites delays the growth of local damage through crack deflection and bridging mechanisms, preventing delamination of the composite layers, and catastrophic failures of components [5].

As an example of a typical assessment experiment, images are captured with synchrotron-based μ-CT , for temperatures between 25 °C and 1600 °C, subjected to different loads. One set of samples consists of rod-like matrices of 1 mm diameter and 55 mm length, reinforced with 500 10 μm diameter Hi-Nicalon fibers, bounded together by a chemical vapor infiltrated SiC matrix. Each fiber is coated with a boron nitride (BN) layer through chemical deposition prior to chemical vapor infiltration of a SiC matrix. The BN layer acts as a weak interface for the matrix cracks to deflect and also facilitates fiber sliding. Fig. 1 shows a composite with the BN layer, which has a lower X-ray absorption coefficient, and therefore often appears as dark rings. Fig. 2 shows the profile and some selected slices of a SiC-based tow with matrix cracks (darker gray).

The 3D representation of the μ-CT is an image stack with high spatial resolution of 0.65 μm per voxel, which favors inspection of structural details at the microscale. Previous work from Bale et al. [1] describes a time-consuming manual process required to extract features from these image samples. This paper reports on the challenges and solutions required to automatically tackle high resolution large datasets of SiC-based composites. As an example of how these tools work, we selected a subset of samples of μ-CT data at room temperature, which are subjected to incremental tensile load by a displacement-controlled loading system. We segment the image stacks using an approach similar to Ushizima et al. [3, 6].

The current work described here focuses on the pattern recognition pipeline (PRP) illustrated in Fig. 3, designed to characterize the specimen structure and, subsequently, track damaged sites in the single tow composite and its fibers; the PRP is described in the next section.

3 Image Analysis Methods

Because fiber cross-sections present similar profiles, particularly after applying F3D filters, a template matching algorithm is well-suited to detect fiber outlines. Template matching uses a prototype model to classify unknown patterns [7] from the image if they resemble each other under a given similarity measure.

First, our proposed PRP requires the user to enter a single μ-CT cross-section with manually identified fibers. The purpose of the user-defined image slice is twofold: it creates a global base for comparison, which we call "Base

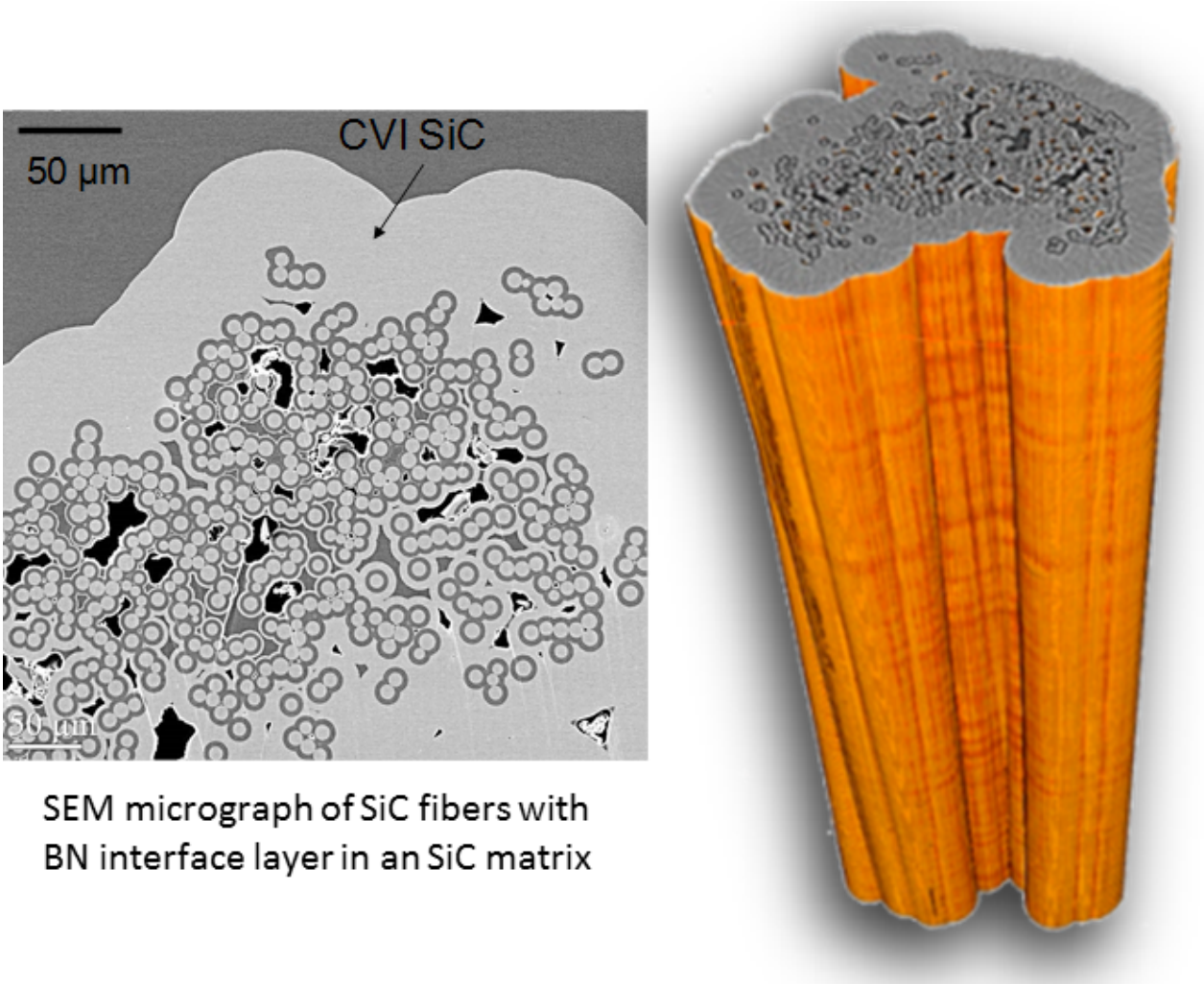


Figure 1: Single tow of SiC fibers and composite matrix

Result" and it also defines a local pattern used to find other individual fiber profiles that appear on slices of the stack. Second, the algorithm calculates the similarity between the prototype and the local region, using the mean squared error (MSE) and cross-correlation coefficient (CC), in order to determine the best matches. This process requires computing the similarity between each pixel (x, y) of the image f and the prototype. In other words, a predefined prototype (template) is centralized at each possible pixel of f to determine the similarity between the template and the overlapping region of f. Notice that the final result of the process is a map of responses for each pixel of f. An empirically calculated tolerance cut-off value determines how much a feature may deviate from the prototype. The tolerance value ranges from 0, meaning exact match, to 100, meaning that the deviation is equal to the variance of the prototype.

The MSE and CC are defined as

$$MSE(x, y) = \frac{1}{n} \sum_{i,j} (p(i, j) - f(x + i, y + j))^2 \tag{1}$$

and

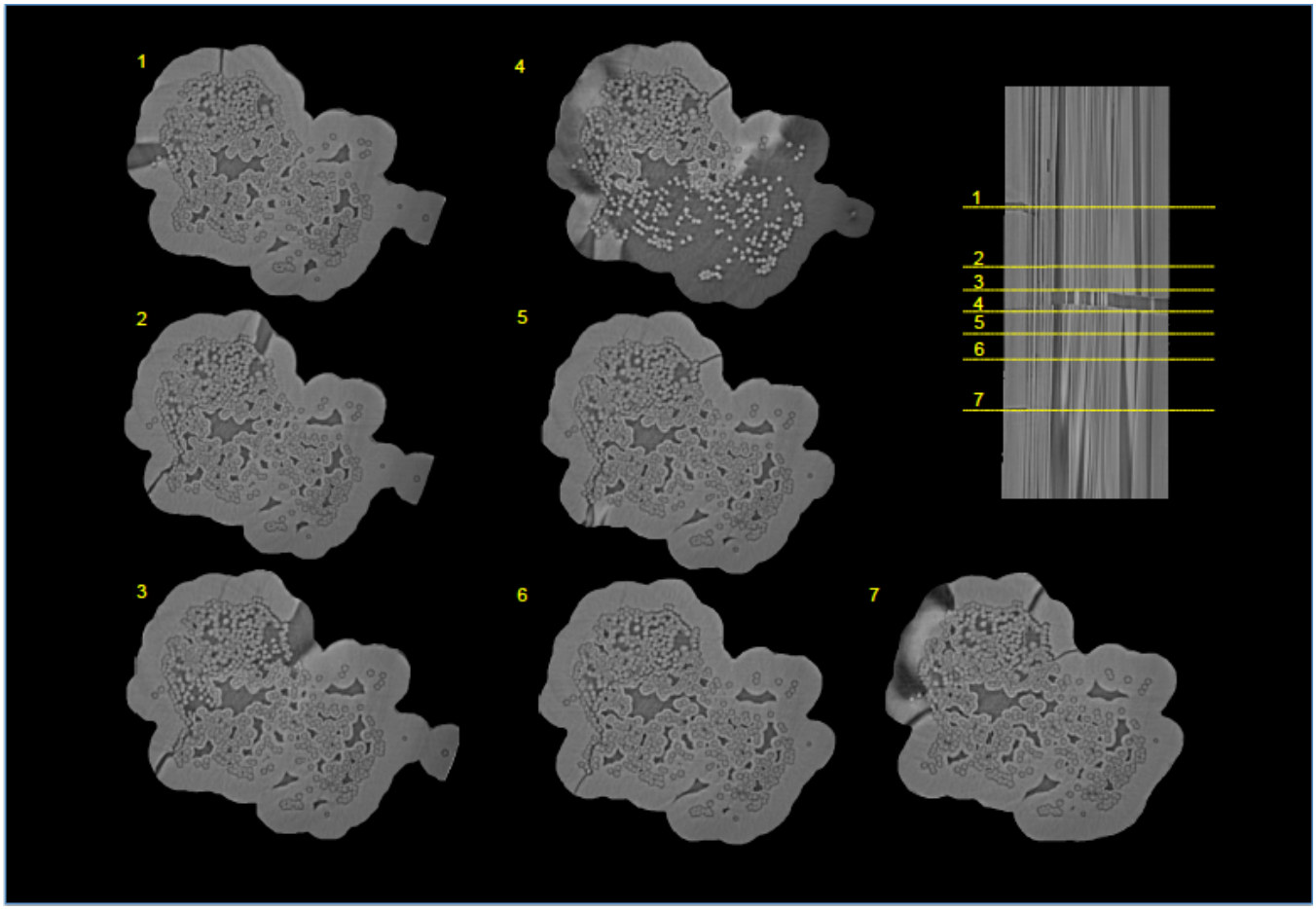


Figure 2: Profile and selected slices of a SiC-based tow with matrix crack. 3D architecture of woven composites impedes the growth of local damage through crack deflection and bridging mechanisms, preventing delamination of the composite layers and catastrophic failures of components.

$$CC(x,y) = \frac{\sum_{i,j}(p_{ij}-\bar{p})\sum_{i,j}(f_{ij}-\bar{f})}{\left[\sum_{i,j}(p_{ij}-\bar{p})^2\sum_{i,j}(f_{ij}-\bar{f})^2)\right]^{\frac{1}{2}}}, \quad (2)$$

where $f_{ij} = f(x+i, y+j)$ for simplicity, $\bar{\cdot}$ is the mean value of $(\cdot)$, and p is the pattern prototype to be found in f. The values of MSE range between $[0,1]$ and the values of CC between $[-1,1]$.

The pipeline in Fig. 3 encloses three main subroutines: preprocessing, template matching and fusion of template matching modes, as described as it follows:

(1) **Preprocessing** of 3D image stacks, by applying bilateral filter and top-hat transform from the proposed F3D plug-in. These functions remove artifacts, such as spurious intensity fluctuations, and enhance separation of the matrix from the composite of the material, allowing better detection of the matrix cracks, and fiber breaks;

(2) **Template matching**, which takes the preprocessed data as input and searches for circular correspondences at each slice, as illustrated by the dashed rectangle in Fig. 3. The template matching runs in two main modes: (a) high tolerance, in which the dissimilarity is allowed to be higher to obtain the maximum amount of positive fibers. This sensitivity is attenuated by including a postprocessing step, which prunes objects unlikely to be part of the fiber network through an intersection with the "Base Result"; (b) low tolerance, which uses prototypes, such as those illustrated at the top right of Fig. 3, to extract candidate pixels with high specificity.

(3) **Fusion of template matching modes** using pixel set operations through two different paths: (a) intersection with "Base Result" and (b) union with the low tolerance template matching result. Notice that our method adapts

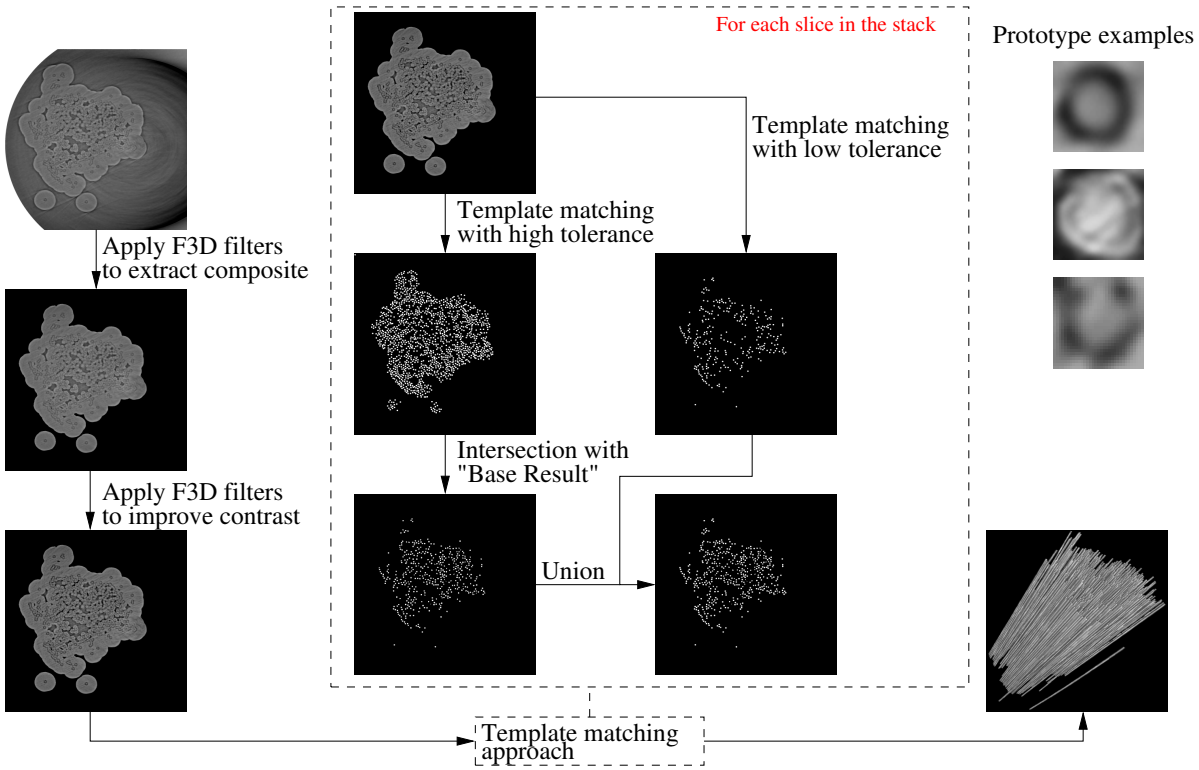


Figure 3: Diagram of the proposed PRP to identify fibers.

to the changes of fiber positions throughout the stack by considering the low tolerance template matching result as an update to the "Base Result" used for the next slice.

The following section describes the F3D plug-in implementation details.

4 F3D: accelerating fiber recovery

Our F3D plug-in contains accelerated key image processing algorithms for fast filtering, and can support segmentation and analysis of high-resolution image datasets. The high resolution imagery requires parallel-capable algorithms that accommodate tens of gigabytes in data sizes with real-time feedback. F3D contains refactored 3D image processing algorithms, such as mathematical morphology operators, bilateral filter [8, 9] and median filter [7] that work as part of Fiji framework.

Our filter kernels are implemented in OpenCL and can be called from our Java code through JOCL [10], which is a Java library that provides Java bindings to OpenCL architecture. F3D gray-level MM operators are one-pass constant time methods that can perform morphological transformations with a line structuring element oriented in several directions. MM operators can be applied to grayscale images, and consist of two parts: (a) a reference shape or structuring element, which is translated over the image, and (b) a mechanism that defines the comparisons to be performed [11] between the image and the structuring element.

Another feature of F3D is the ability to generate formatted instructions that permit using the plug-in as part of Fiji macro-scripts. The formatted instructions are based on JSON [12], which is a JavaScript Object Notation for representing strings with name/value pairs. This way, every process executed by F3D has a corresponding JSON string that is automatically parsed, making it easy to use/reuse F3D-based in Fiji macros.

We compared the performance of the proposed filters to those available in the well-known Fiji 3D Toolkit for mathematical morphology [13]. This section discusses the challenges associated with using existing packages, details about the proposed F3D implementation, and information about how our algorithms promote model-based filtering to recover tubular structures.

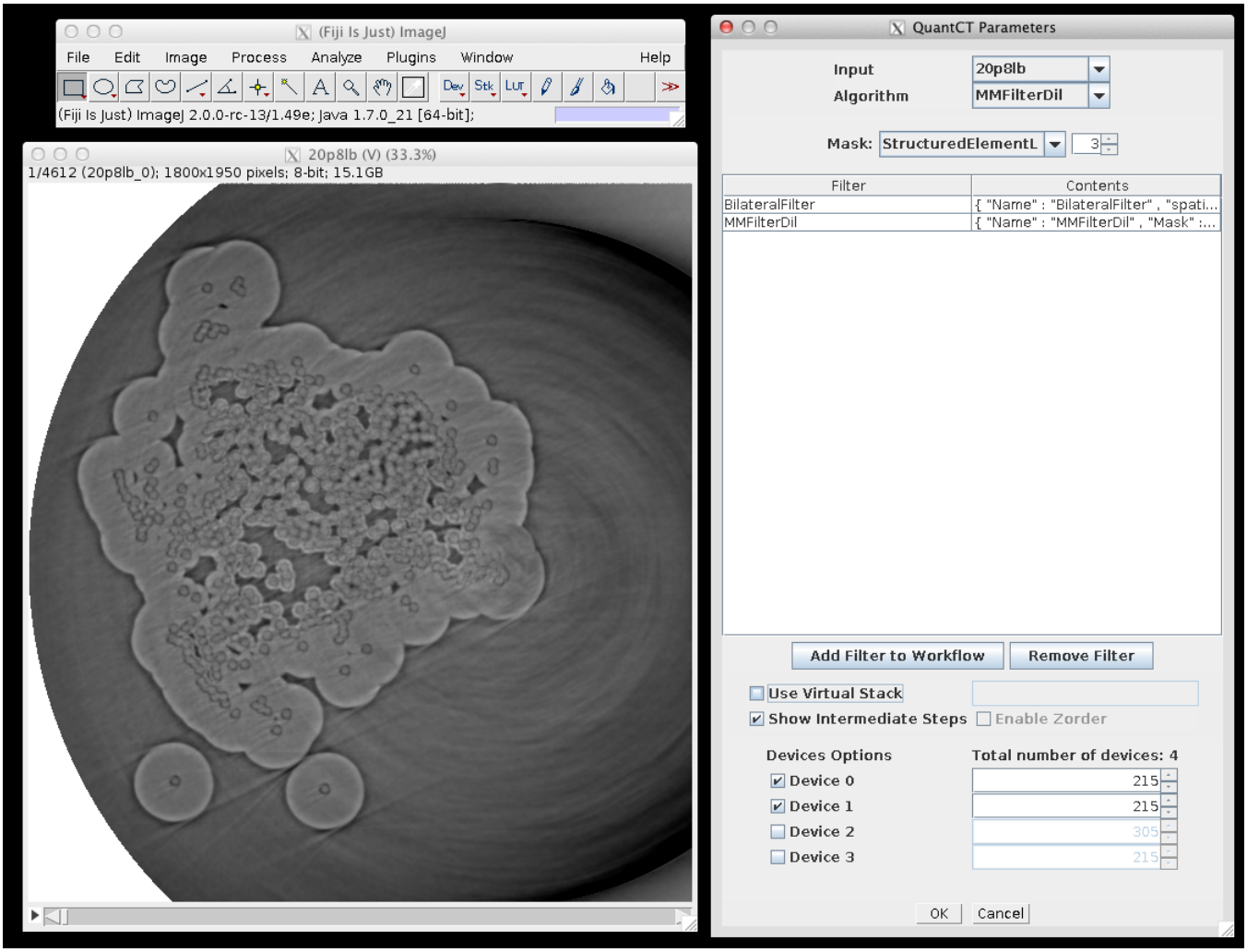


Figure 4: Graphical user interface for F3D plug-in: general purpose image processing tools using multi-GPU.

4.1 Content-based structuring elements

Locally, fibers can be represented by small cylinders. This assumption allows us to control the shape and size expected in a sub-image. We call this task content-based processing because it uses domain information about the objects to preserve and enhance patterns known a priori.

Such small cylinders work as structuring elements (*strel*); similar to probes, they are the means of examining a given image for specific properties, here, tubular structures. The *strel* defines which voxels in the input image will be included in the calculation of the output value. Generally, in gray scale morphology, a *strel* will belong to 2 categories: flat and nonflat, with flat intensity profile and parabolic intensity profile, respectively. Unless mentioned otherwise, the *strel* is symmetrical with flat unit height, and origin at the center [7], which is the implementation available in most of the image processing tools. In contrast, F3D plug-in enables the user to change these common assumptions, particularly allowing the construction of *strel* that presents asymmetry and tridimensionality. The geometric definition of the *strel* is translation invariant in F3D, and we refer to the algorithms in Hedberg et al. [14] and Pinoli et al. [15] for methods that support spatially-variant structuring elements for binary and intensity-based approaches, respectively.

Algorithm 1: Algorithm for F3D OpenCL accelerated image processing.

```
for each OpenCL device do /* threading                                                    */
    Find MaxProcPower for each OpenCLDevice()
    Find MaxNeighbourhood needed over all OpenCL kernels in a pipeline
    while slices available from Input Fiji do
        MSFOCLD [1] ← AvailableMemory/(imageWidth *imageHeight) - MaxOverlap)
        RequestSlices(Min(MSFOCLD, SlicesAvailable))
        Copy Fiji Sub Image Slices to OpenCL InputBuffer
        for w in workflow() do
            if Mask | Bil | Med then
                OpenCLKernel(in, out)
                in ← out
            end
            if Ero | Dil | Open | Close then
                for each structure element do
                    if first structure then
                        OpenCLKernel(in, out)
                        inter ← out
                    end
                    if second to last structure then
                        OpenCLKernel(in, inter, out)
                        inter ← out
                    end
                end
            end
        end
        Add OutputBuffer back of SubSet Slices to ImageStack Buffer hashed by Start and End Slice (memory or disk)
    end
end
Reorder ImageStacks and Return Final Image back to Fiji
```

4.2 Detecting local-linear structures

One of the first algorithms to compute horizontal, vertical and diagonal *strel* for extraction of linear image structures was van Herk's method [16, 17]. He proposed an efficient implementation of erosions and dilations of gray-scale images using only three min/max operations per pixel from 2D images. The optimization comes from dividing the image into blocks of size k, where k is the length of *strel*. It also requires two temporary buffers of the same length as the input image. Limitations include requiring that the size of the buffers be a multiple of k and padding the end of the buffer with negative numbers. Soille et al. [17] generalizes van Herk's algorithm to alleviate some of these limitations, and to allow lines at arbitrary angles. The main contributions in [17] consist of a periodic line to define arbitrary directions and a new recursive algorithm to calculate erosions and dilations, with complexity independent of the k. Both approaches explore the efficiency gained by avoiding unnecessary computer cycles, however they instead rely on memory utilization. This strategy is suitable for small images, but unfeasible when the image size goes beyond half of the RAM available to the computation.

New extensions to those algorithms applied to filamentous structures consider the variation of the structuring elements [14, 15], the combination of MM with Hessian matrix analysis [18], and optimally designed algorithms to applications in biology as in [19, 20]. Similarly, our paper focuses on the optimization of MM operators for fibrillar structures embedded in large image stacks (3D), considering hybrid architectures for fast processing. Differently, our paper explores the capability of graphics cards to perform computation, and minimize data transfers, which enables the proposed PRP to detect fiber breaks and composite cracks from large image files, typically obtained from high resolution μ-CT.

4.3 Prior knowledge into operators

F3D MM operators can use customized structuring elements for spatially-invariant operations, and require: (a) the user-input of the tube diameter in order to define the *strel*; (b) user-defined minimum length inside the tubular structures for piecewise constant formulations. Extending the definitions from previous work [11, 7] to 3-D images, let f be an image defined on the Euclidean space $\mathbb{Z}^3 (f : \mathbb{Z}^3 \to \mathbb{Z})$, and $B \subseteq \mathbb{Z}^3$ a flat or non-flat structuring element and $f(\vec{x} - \vec{b})$ the translation of $f(x)$ by $\vec{b}$ in $\mathbb{Z}^3$. We obtain the erosion ($\ominus$), analogously the dilation, of an image f through a *strel* B by using the following equation: $[f \ominus B](x, y, z) = \min_{(b_x, b_y, b_z) \in B} f(x + b_x, y + b_y, z + b_z)$. We also implemented higher complexity operators such as opening, closing, top-hats, and others by combining the previous operators.

4.4 The OpenCL code behind F3D

Many image processing algorithms are ideally suited for data parallel execution. In this work, we evaluate OpenCL as our deployment platform for image processing, and check its viability for our intra-socket parallelism needs. We implement highly requested image processing algorithms for 3D images as bilateral filter, median filter and morphological operators with variable structuring elements, using OpenCL to capitalize on opportunities to explore data parallel paradigms. Algorithm 1 illustrates the computation under our OpenCL accelerated image processing tools. Our library seamlessly interfaces with Fiji by utilizing JOCL.

Because the Fiji platform provides a data structure called *ImageStack*, which points to a stack of slices, we can use it to represent the final output image without multiple copies of the input data. This data-structure along with hashed reference to the starting index allows for out of order processing, minimizing data movement.

The main strategy is: as long as the target device is able to load at least one frame (subset of slices) along with any dependent frames, then the rest of the dataset can be streamed in that way. This streaming configuration allows our infrastructure to load from out-of-core sources. With that, we can process arbitrarily large datasets in a parallel multi-accelerator environment, removing RAM-based constraints. This capability comes at a cost of latency; our algorithms scale to full-resolution image stacks, larger than RAM, but will increase communication overhead in such situations. The total number of frames streamed at once into the graphics card is currently dependent on the total amount of memory available on each target OpenCL accelerator. By default, our algorithm currently takes one frame as the smallest input. However, our implementation of the image processing OpenCL kernels can be easily extended to deal with further reduced sub-images (portion of an image slice).

To handle multiple accelerators on any given node, a dedicated thread is assigned to each OpenCL device. Each thread requests unprocessed slices up to the maximum allowed by the device constraints (M) automatically, by calculating: $M = maxMem/(w * h) - v/2.0)$, where $maxMem$ is the maximum available memory, w is the image width, h is the image height and v is the maximum overlap. This strategy prioritizes more powerful OpenCL devices to receive a greater subset of image slices. Since this leads to slices potentially being written out of order, we keep track of the processed ranges of images and reorder at the end.

In addition, we also compute an initial heuristic of the processing power of each device using the following formula:$MaxComputeUnits*MaxClockFrequency*MaxMemAllocSize$. If an OpenCL-device is significantly slower than the rest, then that device is removed from consideration. Often these devices are either older generation graphics cards, onboard devices, or a host CPU device on a GPU cluster. We have found that including slower devices significantly degrades overall performance of our library.

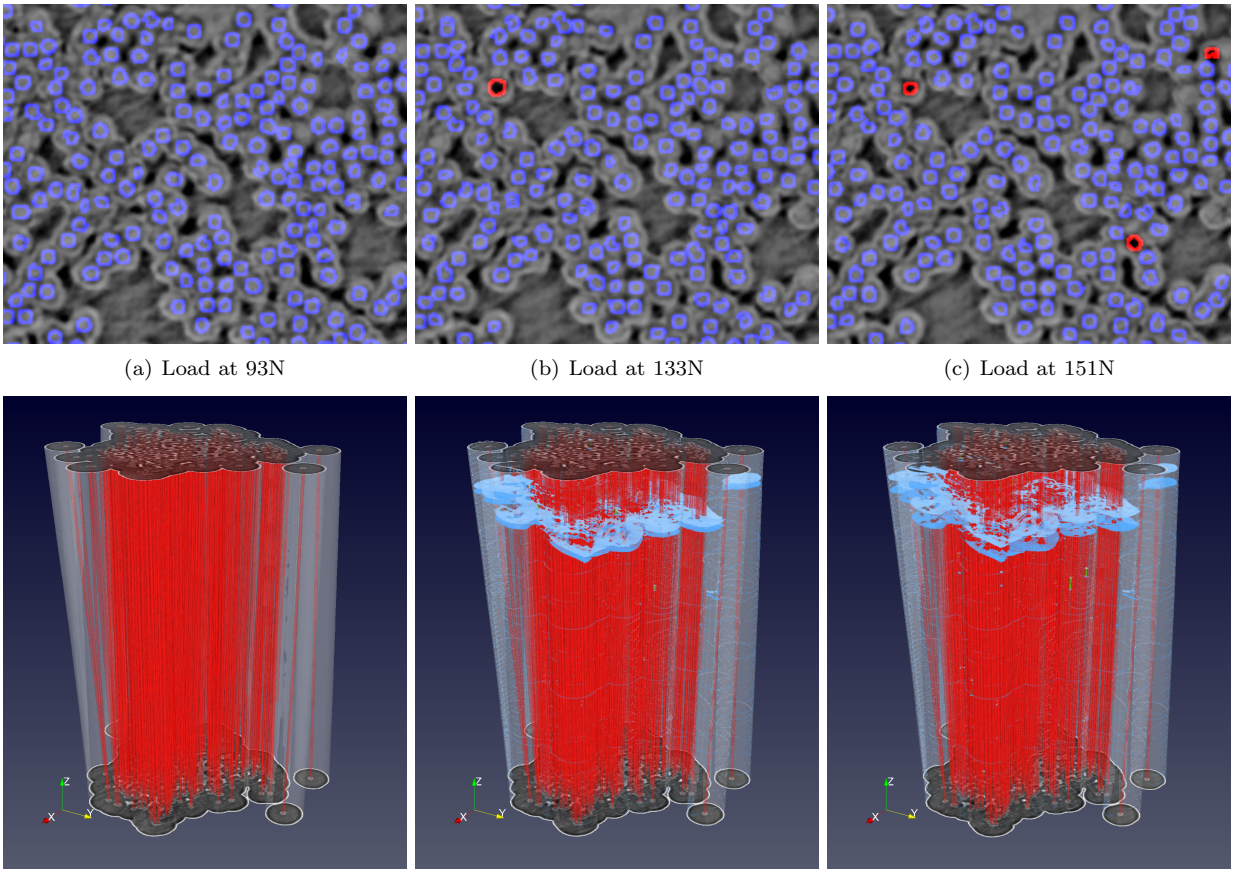
4.5 Benefits of keeping data in memory

Additionally, we designed a sub-pipeline that enables users running consecutive OpenCL accelerated image processing operations to process 3D images in an efficient way. This capability hides the cost of data movement, significantly expedites processing, and reduces costs associated with dynamic compilation and resource allocation. The designed sub-pipeline interface provides two features in order to run efficiently. First, each kernel provides a constraint of the neighbourhood needed to function properly. This allows each sub-pipeline to allocate an appropriate buffer size ahead of the pipeline execution. Second, an input and output buffer, and sometimes a third buffer for kernels that provide max—min operations are created for pipeline execution. Regarding large, out-of-core support using multi-devices, our strategy is to replicate the pipeline across the number of devices and partition the data to stream smaller datasets to each OpenCL-device. In the end we can accumulate the result from each device and either return a final Fiji ImagePlus object or directly write to disk. This enables processing of arbitrarily large datasets (tested: 0.1TB/stack) which often neither fit in the OpenCL-device memory nor in CPU memory.

The complexity of the proposed software architecture is given by:

$$O(a * (n + o)) + b * (n + o) * (\sum_{i=0}^{b}(v_i)) \tag{3}$$

where n is the total number of slices for an input image, o is the maximum overlap over all the algorithms in the sub-pipeline, a is the kernel that operate once over all pixels (e.g. bilateral filter), b refers to the kernels that require performing max—min operations (MM operators) and v is the amount of variable structures for each kernel that supports variable structures.



(a) Load at 93N (b) Load at 133N (c) Load at 151N

(d) No cracks (e) Planar crack at 133N (f) Planar crack + COD at 151N

Figure 5: SiC-based composite with matrix and fibers of a single tow. Top: detected fibers (blue) and fiber breaks (red) using the proposed approach. Bottom: volume rendering of the three SiC composites at different applied loads showing detected fibers (red) and detected matrix cracks (blue) that open up with increasing load along with subsequent fracture of individual fibers (green).

5 Results

5.1 Quantifying damage in μ-CT using the proposed PRP

We run the fiber recognition pipeline shown in Fig. 3 to identify microstructure variations of ceramic composited samples, which were imaged at nine increasing loads in order to allow quantification of the failure behavior of the material. Our PRP relies on F3D filters to enhance matrix cracks and fibers contrast. The recognition algorithms search for composite cracks and fiber breaks to boost applications of fiber-reinforced ceramic composites.

In order to find the composite cracks, we minimize spurious image variations, such as streaks, using F3D bilateral filter with the following parameters: photometric = 20, and geometric = 3, which are associated to low-pass filtering and spatial reachness, respectively. The photometric parameter refers to the standard deviation for intensities in [0,255] within the material, and the geometric parameter corresponds to 5μm, the fiber radius.

Next, we calculate intensity variations between slices using absolute differences to detect sharp changes that are often associated to crack boundaries. Fiber breaks occurred simultaneously with matrix cracks most of the time, although they can also happen out of the crack due to the sliding property of the fibers, as seen in Fig. 5 where three image stacks from our 9-stack dataset are illustrated. Fig. 5(a)-5(c) present a cross-sectional slice taken from three image stacks collected at increasing loads, in which we overlay the contour of the detected fibers in blue and the detected fiber breaks in red. Fig. 5(d)-5(f) present the 3D renderings of the corresponding three stacks showing the detected fibers (in red), the boundaries of the matrix cracks (in blue) and the fibers breaks (in green). The

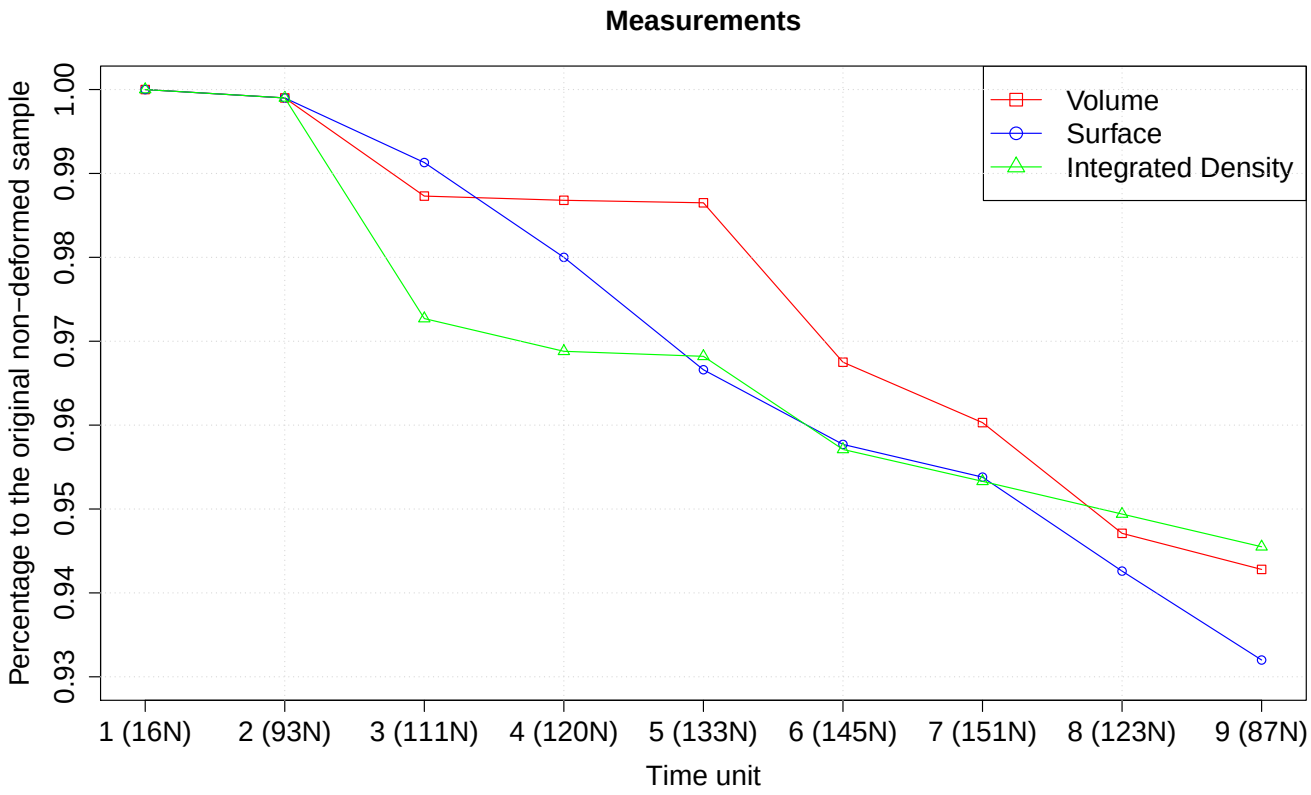


Figure 6: Measurement decay during tensile test experiments: fiber measurements are relative to the original non-deformed sample, subjected to 9 different loads at 25 °C.

increasing of the crack opening displacement with increasing tensile load is captured by the PRP proposed technique (from 110 μm at 133N to 143 μm at 151N), as we emphasize in Fig.4.d-f.

In order to compare the ceramic composite sample deformation according to the tensile forces, we selected three fiber descriptors: volume, surface area and integrated density, the later being the sum of all fiber slices area multiplied by the mean gray value. Fig. 6 presents the evolution of such three measures, obtained from the detected fibers, and ordered by loads. As the load increases over time, the image stacks show a decrease on both volume and surface area with respect to the set of fibers detected. Such changes arise due to the material alteration subjected to the fixed field of view. Deformations include fiber sliding, facilitated by the BN coating layer, and increasing fiber breaks and fiber pull-out distances. With a load of 133N, there are five fiber breaks in the analyzed sample with average pull-out distance of 15 μm, which represents an increase by 2.8x on average. After increasing the load to 151N, the amount of fiber breaks increase 6x with average pull-out distance of 39 μm.

In order to evaluate the accuracy of the proposed PRP, we collected manually segmented slices from material scientists; these substacks are the ground-truth. Using the number of correctly detected voxels as the objective function, we constructed Table 1, which shows the values of Precision and Recall during fiber voxel detection while exposing the sample to nine different loads.

5.2 Performance analysis

As a starting point for filtering composite samples, we considered the grayscale MM operators, developed by Sacha [13], which required a running time of more than an hour for a typical dataset. It would be problematic to deal with the 30GB μ-CT stacks, a typical file size for such imaging experiments. F3D plugin performs such an operation in 2.7 minutes, allowing a significant increase in the size and amount of datasets to be processed, particularly taking advantage of High Performance Computing (HPC) environments that uses GPU. In addition, most modern devices such as Graphics Cards, Modern CPUs, and Xeon Phi provide OpenCL abstractions, which we expect to allow F3D

Table 1: Correctly detected voxels of fibers: quantitative evaluation of the fiber segmentation using proposed PRP against provided ground-truth.

Data	Precision	Recall
16N	0.9215	0.9586
93N	0.95	0.98
111N	0.9139	0.9625
120N	0.9379	0.9418
133N	0.9454	0.9753
144N	0.9483	0.984
151N	0.9454	0.9726
122N	0.9468	0.9746
87N	0.9442	0.971

code to be written once and executed on many different modern computing environments.

F3D implementation runs tens of times faster than the plugin in [13] when graphics cards are used. Our performance results consider a Intel(R) Xeon(R) CPU E5-2660 at 2.20GHZ, 62GB RAM, 3 NVIDIA Tesla K20X and 1 K40m GPU. We compare the run time of the filters in Fig. 7, where the green line corresponds to Sacha's plugin, the red line represents F3D and the blue line represents F3D using virtual stack. The speed improvement is 17x in average (30Gb dataset) in comparison with Sacha's plugin. Additionaly, F3D was able to process a dataset of 100GB, more than the system available RAM, in approximately 40 minutes using the virtual stack.

6 Discussion

SiC-based ceramic composites have potential applications in gas turbines and hypersonic flight aircraft design, and microstructure characterization is fundamental to evaluate how loads and temperature variations impact the material. We introduced a recognition pipeline based on structure enhancing filters from F3D and template matching for detecting fibers woven into ceramic composities. F3D filters were also fundamental in allowing crack detection through slice differentiation. We presented promising results on the application of the proposed PRP to nine different μ-CT stacks. Initially, one of the bottlenecks of the PRP was the filtering procedure took more than one hour for high resolution datasets (30GB) using available packages [13], besides its inability to use user-defined structuring elements. Our F3D plugin solves these problems, and combines OpenCL parallel code with Java to deliver optimized operators that run in near real-time, specifically 2.7 min for image files.

The current bottleneck in our proposed pipeline is the 2D template matching process, which is applied two times for each slice of the stack independently. This will be improved by transforming the two template matching modes into a set of highly parallel processes. Given that the μ-CT stacks have almost $5,000$ slices, the use of more than one "Base Result" per stack, dividing it into different substacks, is adequate and feasible. Consequently, the template matching process can be applied in parallel for each "Base Result", i.e. for each substack created.

7 Conclusion and Future Work

Our quantitative analysis methods were tailored to characterize both the structure of a given material system and the evolving damage in corresponding datasets collected during in-situ μ-CT experiments. This paper describes the problem of identifying micro-damage, and the design of a pattern recognition pipeline to detect cracks and fiber breaks from materials imaged by μ-CT. Such pipelines required the development of fast filters that can use GPU for near real-time processing of collected samples. F3D implements graphic-card aware algorithms that are quite common in μ-CT volume analyses, which otherwise would be infeasible through serial computations dependent on the image size.

Future directions include additional automated inspections of the μ-CT data, such as the extraction of the crack opening displacement, statistics across the stack, classification of cracks into its sub-groups such as planar, spiral and branching, and to extend this study to more composite samples, and at different temperatures. We also intend to enable Z-order support, which has been shown to operate extremely efficiently for algorithms that

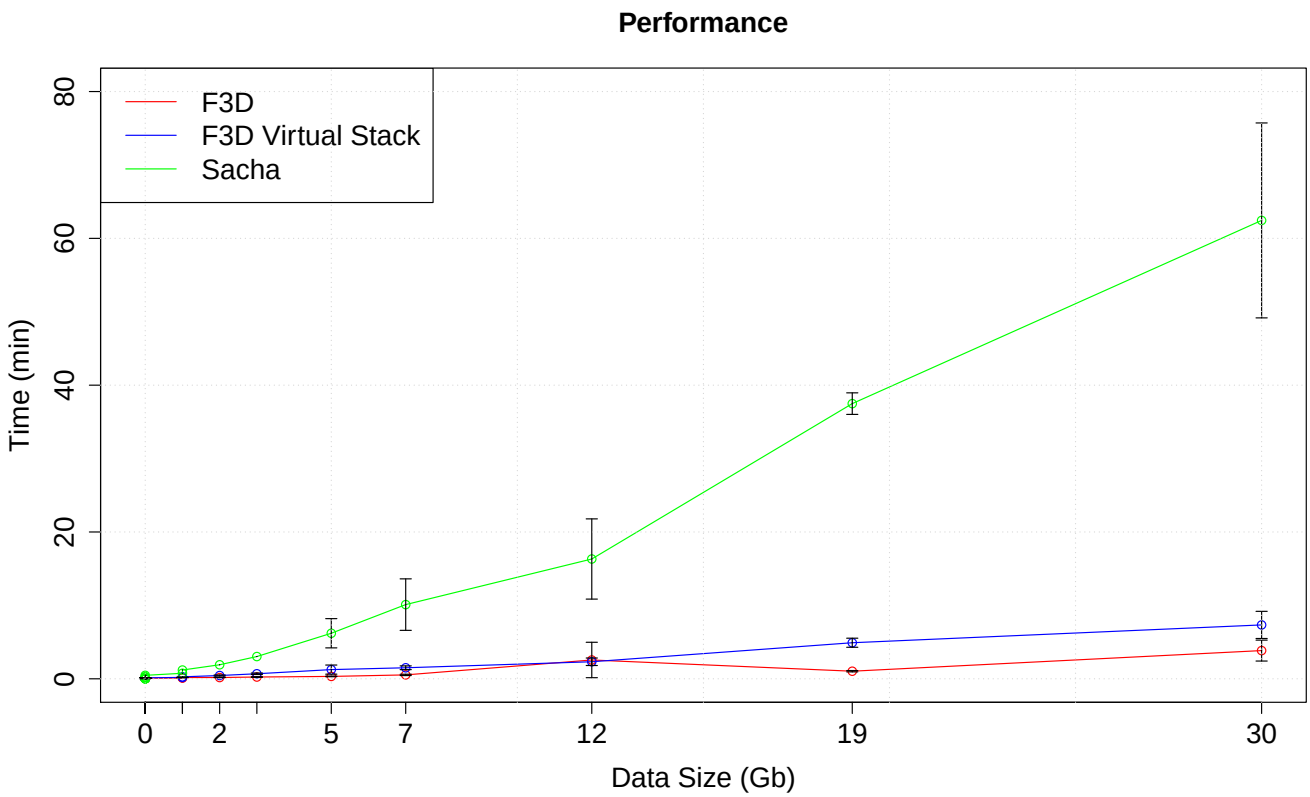


Figure 7: Performance comparison of F3D with Sasha's [13] plug-in for different dataset sizes: each point represent the average run-time of 5 trials.

perform neighborhood lookups, which is the case of the filters in F3D. Another potential exercise is to test our library over multiple nodes of a HPC environment using a Map-Reduce infrastructure. Further investigations will include parallelization of template matching modes, including new image representations for improving speed up of this step [21], for example, by using integral images [22] for rectangular based prototypes.

Acknowledgment

This work was supported by the Director, Office of Science, Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Our project is part of the Center for Applied Mathematics for Energy Research Applications (CAMERA), hosted at Lawrence Berkeley National Laboratory. We would like to thank Abhinav Sarje for sharing the GPU infrastructure with our team, and NERSC for the support on generating high resolution visualizations.

References

[1] H. A. Bale, A. Haboub, A. A. Macdowell, J. R. Nasiatka, D. Y. Parkinson, B. N. Cox, D. B. Marshall, and R. O. Ritchie, "Real-time quantitative imaging of failure events in materials under load at temperatures above 1,600C." *Nat Mater*, vol. 12, pp. 40–46, 2012.

[2] M. N. Rossol, J. Shaw, H. Bale, R. O. Ritchie, D. B. Marshall, and F. W. Zok, "Characterizing weave geometry in textile ceramic composites using digital image correlation," *Journal of The American Ceramic Society*, vol. 96, no. 8, pp. 2362–2365, 2013.

[3] D. M. Ushizima, D. Morozov, G. H. Weber, A. G. Bianchi, J. A. Sethian, and E. W. Bethel, "Augmented topological descriptors of pore networks for material science," *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Vis 2012)*, vol. 18, no. 12, pp. 2041–2050, 2012, lBNL-5964E.

[4] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, "Fiji: an open-source platform for biological-image analysis," *Nat Meth*, vol. 9, no. 7, pp. 676–682, Jul. 2012.

[5] D. B. Marshall and B. N. Cox, "Integral textile ceramic structures," *Annual Review of Materials Research*, vol. 38, pp. 425–443, 2008.

[6] D. M. Ushizima, A. Bianchi, C. deBianchi, and E. W. Bethel, "Material Science Image Analysis using Quant-CT in ImageJ," in *Proceedings of ImageJ User and Development Conference*, Oct. 2012.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–846.

[9] E. W. Bethel, "Exploration of Optimization Options for Increasing Performance of a GPU Implementation of a Three-Dimensional Bilateral Filter," LBNL, Berkeley, CA, USA, 94720, Tech. Rep. LBNL-5406E, 2012.

[10] "Java bidings for OpenCL," http://www.jocl.org/, 2009, [Online; accessed 13-Jul-2014].

[11] M. V. Droogenbroeck and H. Talbot, "Fast computation of morphological operations with arbitrary structuring elements," *Pattern Recogn Lett*, vol. 17, pp. 1451–1460, Sep. 1996.

[12] "JSON - JavaScript Object Notation," http://json.org/, 2013, [Online; accessed 10-Sep-2014].

[13] J. Sacha, http://ij-plugins.sourceforge.net/plugins/3d-toolkit/, 2013, [Online; accessed 13-Jul-2014].

[14] H. Hedberg, P. Dokladal, and V. Owall, "Binary morphology with spatially variantstructuring elements: Algorithm and architecture," *IEEE Trans Image Process*, vol. 18, no. 3, Mar. 2009.

[15] J. C. Pinoli and J. Debayle, "Spatially and intensity adaptive morphology," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 7, pp. 820–829, 2012, special Issue on Filtering and Segmentation in Mathematical Morphology.

[16] M. van Herk, "A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels." *Pattern Recognition Letters*, vol. 13, no. 7, pp. 517–521, 1992.

[17] P. Soille, E. J. Breen, and R. Jones, "Recursive implementation of erosions and dilations along discrete lines at arbitrary angles," *IEEE Trans PAMI*, vol. 18, no. 5, pp. 562–567, May 1996.

[18] A. Dufour, O. Tankyevych, B. Naegel, H. Talbot, C. Ronse, J. Baruthio, P. Dokládal, and N. Passat, "Filtering and segmentation of 3d angiographic data: Advances based on mathematical morphology," *Med Image Anal*, vol. 17, no. 2, pp. 147–164, 2013.

[19] G. Herberich, R. Windoffer, R. Leube, and T. Aach, "3D segmentation of keratin intermediate filaments in confocal laser scanning microscopy," *IEEE EMBS*, pp. 7751–7754, 2011.

[20] G. Herberich, A. Friedrich, T. Aach, R. Windoffer, and R. Leube, "Flux-based 3d segmentation of keratin intermediate filaments in confocal laser scanning microscopy," *IEEE ISBI*, pp. 166–169, 2012.

[21] S. Korman, D. Reichman, G. Tsur, and S. Avidan, "Fast-match: Fast affine template matching," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 1940–1947.

[22] T. Wu and A. Toet, "Speed-up Template Matching through Integral Image based Weak Classiers," *Journal of Pattern Recogn Research*, vol. 9, no. 1, pp. 1–12, Jan. 2014.