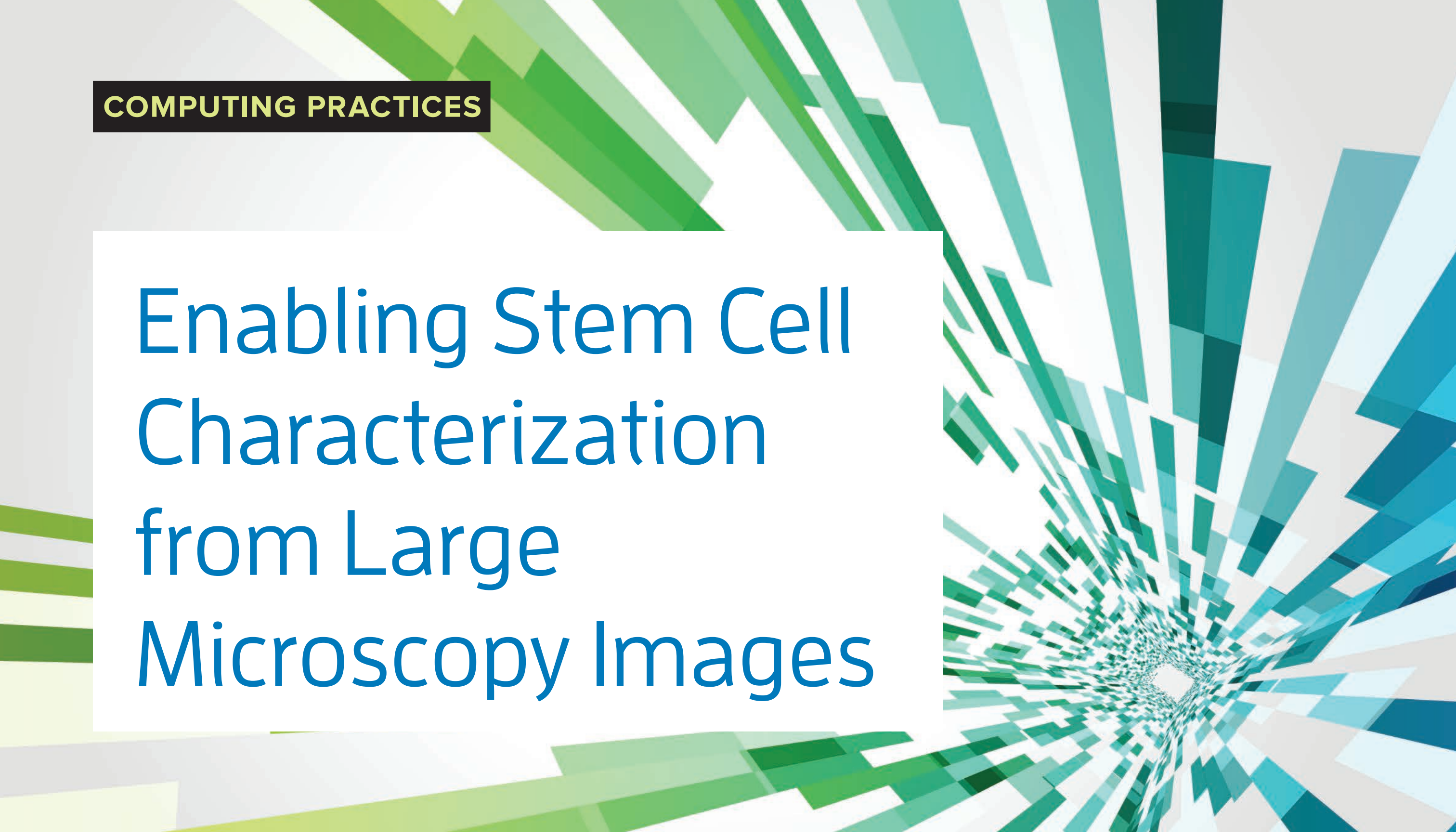


Enabling Stem Cell Characterization from Large Microscopy Images

Peter Bajcsy, Antoine Vandecreme, Julien Amelot, Joe Chalfoun, Michael Majurski, and Mary Brady, National Institute of Standards and Technology

Microscopes can now cover large spatial areas and capture stem cell behavior over time. However, without discovering statistically reliable quantitative stem cell quality measures, products cannot be released to market. A Web-based measurement system overcomes desktop limitations by leveraging cloud and cluster computing for offline computations and by using Deep Zoom extensions for interactive viewing and measurement.

Biomanufacturing stem cell therapies holds great promise for healthcare, but the clinical use of stem cell products requires quality measurements that capture stem cell populations' dynamic behavior. Without high confidence in such quality measurements, products cannot move from trials to market. For example, as of mid-June 2016, 612 mesenchymal stem cell (MSC) clinical trials have been completed or are ongoing, yet no MSC-based products have reached the market (https://clinicaltrials.gov/ct2/results?term=mesenchymal+stem+cell&Search=Search).

Image size is a formidable challenge in obtaining quality measurements, as one field of view (FOV) for a microscope represents only 0.0626 percent of the spatial area for a circular region with 3.494-cm diameter imaged at 10× magnification. Spatially stitching megapixel FOV images yields one gigapixel 2D image for each timepoint and image modality. A stack of gigapixel image frames over five days forms a Tbyte 3D volume with spatial [*x*, *y*], time [*t*], and image modality dimensions.

Although microscopy imaging technology is available, scientists have no off-the-shelf solution with which to interactively inspect these volumes and create subsets to measure and analyze. The computationally intensive preprocessing steps—image calibration, stitching, segmentation, feature extraction, and modeling—far outpace desktop computing's RAM capacity, which means that Tbyte images cannot be loaded and scientists cannot interactively explore them.

The current workaround is to capture stem cell images at low resolution with a few high-resolution,

0018-9162/16/$33.00 © 2016 IEEE

CHALLENGES IN MOVING TO LARGE MICROSCOPY

The move to large microscopy coverage has encountered three main roadblocks: image-collection automation, collaborative work, and data sharing and interactive measurement. These three roadblocks often give scientists pause when faced with measuring Tbyte images.

IMAGE-COLLECTION AUTOMATION

Automating microscopy acquisition involves collecting a large number of fields of view (FOVs) using several spectral bands and instrument modalities over extended time periods as well as automating image-calibration and image-quantification steps.

COLLABORATIVE WORK

In a collaborative research environment, scientific collaborators must share a large 3D volume, which requires storage on a network-accessible server because volume size if far greater than the RAM of a single desktop or laptop can handle. The challenge is how to transition from single desktop computations to commercially available cloud computing environments in a way that accommodates data size and computational requirements.

DATA SHARING AND INTERACTIVE MEASUREMENTS

Finally, there must be some way to share image data and interactively measure images of unprecedented specimen coverage. Given the large number of image files, multiple experts cannot rapidly inspect the data holistically using the raw image files. After assembling raw image tiles into a large 2D image frame, the composite image dimensions are much larger than a typical computer screen. Thus, inspection tools must extend Deep Zoom technology by enabling multiple zoom levels and panning that supports 3D navigation.

Measurements must be taken to form enough images to give scientific findings sufficient statistical significance, ensure that measurements are complete, and enable the identification of rare but significant events. Current open source solutions do not provide direct quantitative measurement capability, lack the accuracy and uncertainty evaluations of the image-processing steps used, and require unprecedented computational resources to enable interactive quantitative measurements.

limited FOV samples. This less-than-ideal approach has led to problems that include errors in characterizing temporal stem cell behavior in terms of cell states (migrating, dividing, differentiating, or dying),[1] very low reproducibility of published work,[2] and inadequate conclusions about how cell states vary. Recent advances in automated acquisition combined with computational image stitching and visualization have enabled high-resolution imaging while retaining a large imaging area, and the stitched 2D images have led to virtual nanoscopy with pan and zoom capabilities.[3] Although virtual nanoscopy is a step forward, it also requires computational capabilities to enable viewing and measurements over collections of Tbyte images.

Microscopy imaging could be a more promising method to characterize cell colonies according to their growth rate and spatial heterogeneity (the distribution of cells that become visible as bright or dark after introducing a biomarker). Both growth rate and spatial heterogeneity are promising quality indicators—but only if quantitative measurements can be collected over multiple spatial and temporal scales. With cells changing states over time and being several orders of magnitude smaller than cell products, modern microscopes must be able to image large spatial areas, repeat imaging over time, and acquire images over several image modalities. They can accomplish all these tasks already, but, as the sidebar

"Challenges in Large Microscopy Coverage" describes, the microscopy-based characterization of stem cell products from so many large images faces several obstacles, primarily the need for interactive inspection that yields statistically reliable quantitative measurements, which, in turn, can serve as the basis for product-release criteria.

To fill that need, we developed a Web-based measurement system consisting of

- offline image-processing algorithms that we redesigned from their desktop versions to run on a computer cluster, and
- extensions to Microsoft's Deep Zoom within a client–server framework (http://isg.nist.gov/deepzoomweb),
- plugins to the OpenSeadragon Web-based viewer (http://openseadragon.github.io) to support the interactive Web-based measurement of stem cell objects, and the selection and downloading of image subregions, or subsets.

To guide scientists in configuring a similar interactive measurement system in application environments, a series of tradeoffs were documented. The tradeoffs include hardware, software, and network configurations and are related to collocating data and computational resources in desktop, cluster, and client–server environments.

REDESIGNING DESKTOP ALGORITHMS

As a first step in redesigning the desktop image-processing algorithms for cluster computing, we collected images from three live stem cell preparations. Most advanced microscopes allow automated image acquisition using bundled proprietary software or open source frameworks such as µManager[4] to control the microscope. Our collected images were acquired by a Zeiss 200M microscope controlled by Zeiss Axiovision software. The software acquired two images for each FOV—corresponding to the two imaging modalities (phase contrast [PC] and green fluorescent protein [GFP]) every 45 minutes over five days.

After assembling the acquired FOVs into a composite image of the stem cell specimen, each frame of a time sequence consists of approximately 23,000 × 21,000 pixels with 16 bits per pixel (bpp). One composite 2D frame represented is a subregion representing approximately 19.08 percent of the area for the round well with 3.494 cm diameter. Loading one 2D frame requires close to 1 Gbyte of RAM. The three stem cell preparations were imaged for 161, 157, and 136 time points yielding 77.8, 75.8, and 65.7 gigapixel volumes. In all, we had 359,568 image files—approximately 0.9 Tbyte.

Image processing

Our processing pipeline consists of flat-field and background correction, stitching, colony segmentation, colony tracking, image-feature extraction, pyramid building, and reprojection. With this many stages, we needed multiple image-processing libraries, and the sheer size of the three 4D volumes made any offline image processing on a desktop extremely time consuming. Moreover, most image-processing libraries—including ImageJ/Fiji, OpenCV, Matlab, and Java advanced imaging—are not written for cluster or cloud computing. Part of the redesign to make desktop algorithms suitable for cluster computing required that we rework them to leverage distributed computational resources, which mandated purchasing licensed software for all the cluster nodes.

The redesigned algorithms are based on either the Hadoop framework[5] or on Java Remote Method Invocation (RMI) with an in-house–designed job scheduler. The algorithms include flat-field correction, segmentation based on convolution kernels, image-feature extraction, and pyramid building. We also used Hadoop's Map and Reduce middleware to parallelize algorithms and Hadoop's various mechanisms for managing hardware failures and monitoring executions.

Most of the algorithms are downloadable (https://isg.nist.gov/deepzoomweb/activities), and Tbyte-sized 3D volume examples from cell biology and materials science are accessible via the prototype interactive Web-based system (http://isg.nist.gov/deepzoomweb).

Efficiency assessment

We redesigned algorithms using Hadoop and Java Remote Method Invocation (RMI) and assessed the algorithms for computational efficiency on both a multicore desktop and varying cluster configurations,[6] calculating efficiency metrics for execution times over an increasing number of nodes. We then ranked the suitability of each redesigned algorithm for cluster computing. Our rankings showed that most of Hadoop-redesigned algorithms outperformed the original implementations using RMI clusters and the multicore desktop.

Our efficiency tests had to accommodate RAM requirements per node, data transmission, data packaging, and I/O tasks. Because of image volume, some of the more taxing tasks were to load a single timeframe without subdividing it, input hundreds of thousands of images to the flat-field correction algorithms, and generate the several millions of images as outputs from pyramid construction. Our test datasets (available at https://isg.nist.gov/deepzoomweb/data) and efficiency benchmarks are designed to help scientists who must rely on large-coverage microscopy transition their image-processing computations from desktop to Hadoop-based cluster computing.

BUILDING A CLIENT–SERVER CONFIGURATION

To provide interactive viewing in the creation and measurement of stem cell image subsets, we needed not only algorithms that would work on cluster computing but also a client–server configuration that would maximize the efficiency of retrieval, transmission, and viewing of large images; have the necessary mechanisms to foster interactivity; and have optimizations to support a variety of scientific applications.

The building blocks for designing and deploying a client–server Web-based system exploit several technologies for hosting images on a server, such as SQLite, MySQL, and Apache Tomcat; for rendering content on the client side, such as OpenSeadragon, D3, and XTK; and for communicating between clients and the server, such as RESTful webservices, HTML5, and JavaScript. The application domain dictates the exact customization, integration and optimization of these technologies. We opted to use open source components, and our client–server design is not tied to any specialized hardware.

Client–server systems for image inspection already exist for a number of domains: Aladin Sky atlas for astronomy images; Collaborative Annotation Toolkit for Massive Amounts of Image Data (CATMAID) for brain images;[7] and the USGS Global Visualization Viewer (GloVis) and NASA's Global Imagery Browse Services (GIBS) with Worldview for satellite images. These systems can handle terapixel-sized datasets, and even the petapixel images encountered in selective-plane illumination and electron microscopy.[8,9] However, at that scale the goal of interactive image inspection requires the use of a high-speed network as well as specialized hardware and software for processing and viewing the images, such as a GPU-based framework for volume ray casting.[9]

CHARACTERIZATION FROM MANY IMAGES REQUIRES INTERACTIVE INSPECTION AND QUANTITATIVE MEASUREMENTS THAT ARE STATISTICALLY RELIABLE.

DEEP ZOOM EXTENSIONS WITH OPENSEADRAGON

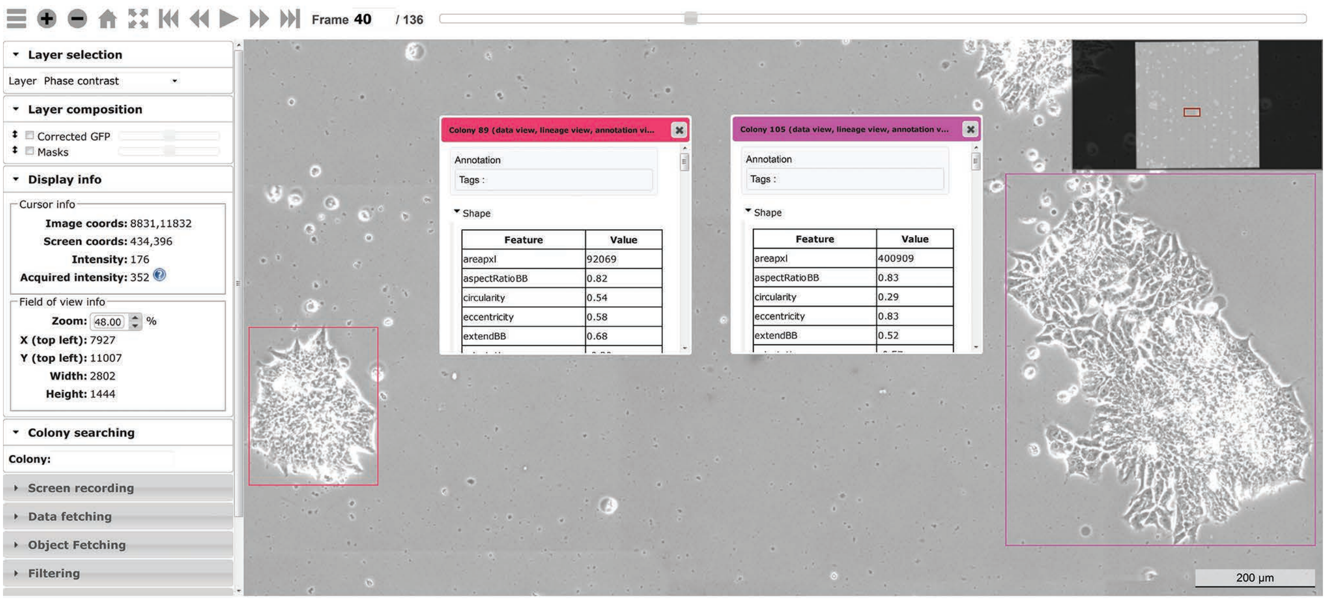
To address both data size and interactivity, we used the OpenSeadragon JavaScript library to extend Deep Zoom–based visualization. We added physical units to make image values meaningful and widgets to support subsetting (specifying subset parameters), intensity and distance measurements and the hyperlinking of spatial statistics with temporal lineage. The OpenSeadragon plugins and measurement widgets are downloadable at the GitHub repository (https://github.com/usnistgov/{OpenSeadragonScalebar, OpenSeadragon-PixelColor, OpenSeadragonFiltering, WebDeepZoomToolkit}).

With extended Deep Zoom, scientists can interactively inspect and measure GFP and PC intensities, their side views (which are orthogonal projections of 3D volumes), background-corrected GFP intensities, and segmentation masks.

Interactive image inspection

Deep Zoom technology uses a multi-resolution representation of each 2D image (which is one time frame) partitioned into tiles of 256 × 256 pixels. It also uses the pyramid representation in supporting the on-demand access of 3D subsets, which can take place during several stages:

- *viewing*, which involves tile access and retrieval;
- *downloading* of regions, which involves tile selection, reconstruction of the requested image area, zip compression, and retrieval; and
- *computing*, which involves tile access, retrieval, and pixel manipulation.



Layer: Phase contrast

□ Corrected GFP
□ Masks

Cursor info
Image coords: 8831,11832
Screen coords: 434,396
Intensity: 176
Acquired intensity: 352

Field of view info
Zoom: 48.00 %
X (top left): 7927
Y (top left): 11007
Width: 2802
Height: 1444

Colony:

Colony 89 (data view, lineage view, annotation vi...

Annotation
Tags :

Shape

Feature	Value
areapxl	92069
aspectRatioBB	0.82
circularity	0.54
eccentricity	0.58
extendBB	0.68

Colony 105 (data view, lineage view, annotation v...

Annotation
Tags :

Shape

Feature	Value
areapxl	400909
aspectRatioBB	0.83
circularity	0.29
eccentricity	0.83
extendBB	0.52

FIGURE 1. Extended Deep Zoom controls for image interactions in a regular browser view. The panels in the upper left corner and the slider at the top provide information about image intensities, zoom level, frame selection, and information layer. The panels floating over the image data provide information about the selected colonies along with links to the data, lineage, and annotation views for those colonies. Additional panels on the left enable colony searching, screen recording (movie player), subsetting, distance measurements, and scale-bar inclusion. This additional information gives scientists a way to interpret the values displayed in a basic Deep Zoom view. The functions in the left panel allow additional collaborative research and annotation for every region of interest.

The downloading and computing functions differ from those in other client–server solutions that use the pyramid representation in that the user specifies the downloaded image's region of interest, colony identification, resolution, and timeframes. All interactive computations are performed on the 8-bpp image tiles that the browser retrieves.

The pyramid representation of these GFP, PC, and mask image layers consists of approximately 11 million files in a half million folders. The files and folders include 8-bpp, 16-bpp, and 32-bpp representations of data from floating-point operations during background correction and calibration (preprocessing steps). Because browsers currently support 8-bpp images, we chose 8 bpp for downloading in an interactive mode, in which scientists view the image with a browser, and reserved 16 bpp and 32 bpp for downloading in a fetching mode, in which scientists specify parameters of raw or processed image subsets. Subsetting enables additional measurements of and research into the images.

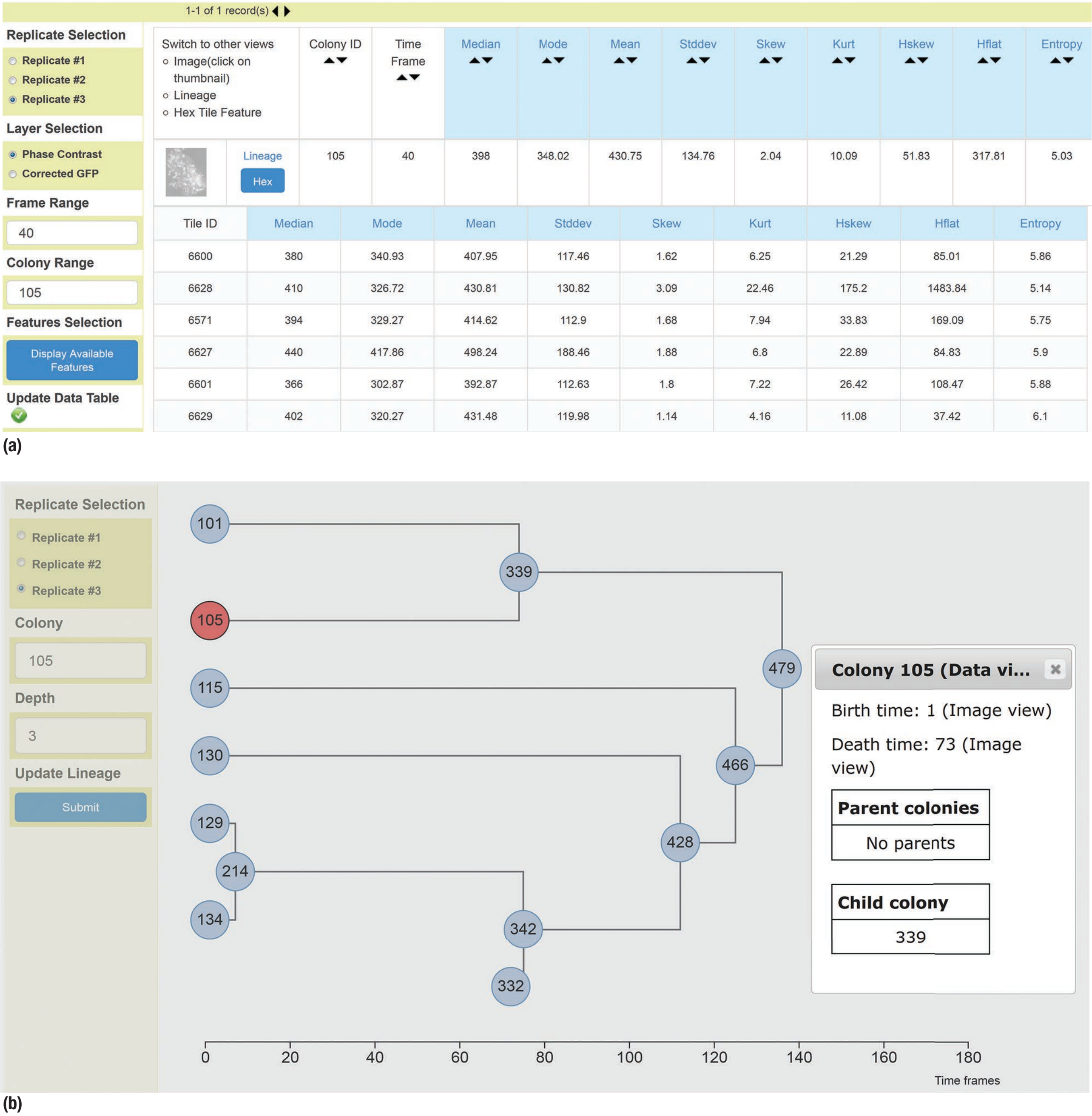
The neighboring contextual image information must be viewed in three orthogonal planes: [*x*, *y*], [*x*, *t*], or [*y*, *t*]. Any other plane not parallel to the three orthogonal planes is considered oblique. The measurements from oblique planes are not useful because they lack meaningful units for interpretation. Figure 1 shows the extended Deep Zoom interface for visual inspection of cells in [*x*, *y*] plane.

Hyperlinking characteristics

Enabling visual image inspection only through a browser is insufficient when determining stem cell characteristics because biomanufacturing involves a high number of stem cell colonies. For example, in our spatial coverage (19 percent of the specimen), we observed about 300 colonies in the initial timepoint, but as the colonies grew and merged over time, we counted more than 1,000 unique time-tracked colonies in each preparation. This contrasts sharply with the labor-intensive process of manually spot-checking colonies at each timepoint.

With this many colonies, the process of extracting and comparing stem cell colonies' intensity, shape, and texture characteristics must be automated to ensure that scientists correctly determine stem cell states at population and individual levels. In our work, we automatically segmented and extracted 75 image characteristics (features) per colony. We also partitioned each colony into hexagonal regions and extracted the same features per hexagon to study spatially local properties. Feature extraction yielded approximately 2.6 Gbytes of data per cell preparation. Finally, we hyperlinked all spatial, temporal, image-modality, and feature information for each colony so that scientists can switch between Web interfaces (various hyperlinked views of stem cell data) to gain additional insights into each stem cell colony under scrutiny.

Figure 2 shows the colony features and temporal lineage information hyperlinked with the image information in Figure 1. Hyperlinked views are central to understanding the spatial, temporal, image-modality, and feature



1-1 of 1 record(s) ◀ ▶

Replicate Selection

○ Replicate #1
○ Replicate #2
● Replicate #3

Layer Selection

● Phase Contrast
○ Corrected GFP

Frame Range

40

Colony Range

105

Features Selection

Display Available Features

Update Data Table ✓

Switch to other views ○ Image(click on thumbnail) ○ Lineage ○ Hex Tile Feature		Colony ID ▲▼	Time Frame ▲▼	Median ▲▼	Mode ▲▼	Mean ▲▼	Stddev ▲▼	Skew ▲▼	Kurt ▲▼	Hskew ▲▼	Hflat ▲▼	Entropy ▲▼
	Lineage Hex	105	40	398	348.02	430.75	134.76	2.04	10.09	51.83	317.81	5.03

Tile ID	Median	Mode	Mean	Stddev	Skew	Kurt	Hskew	Hflat	Entropy
6600	380	340.93	407.95	117.46	1.62	6.25	21.29	85.01	5.86
6628	410	326.72	430.81	130.82	3.09	22.46	175.2	1483.84	5.14
6571	394	329.27	414.62	112.9	1.68	7.94	33.83	169.09	5.75
6627	440	417.86	498.24	188.46	1.88	6.8	22.89	84.83	5.9
6601	366	302.87	392.87	112.63	1.8	7.22	26.42	108.47	5.88
6629	402	320.27	431.48	119.98	1.14	4.16	11.08	37.42	6.1

(a)

Replicate Selection

○ Replicate #1
○ Replicate #2
● Replicate #3

Colony

105

Depth

3

Update Lineage

Submit

Colony 105 (Data vi... ✖

Birth time: 1 (Image view)

Death time: 73 (Image view)

Parent colonies
No parents

Child colony
339

(b)

FIGURE 2. Hyperlinked views for colony 105's image information in Figure 1. (a) Tabular data view of colony 105's features, which were computed on the hexagonal partitioning (tile ID) of colony 105 at frame point 40. (b) Lineage view of colony 105 showing its temporal connections to child or parent colonies resulting from the colony merger. Hyperlinking the image, data, and lineage views enables explorations such as interactive subsetting by colony ID and frame ID. Results can then be displayed with the requested tabular feature information and downloaded as comma-separated value (CSV) files.

aspects of big data. Within each view, a scientist can reconfigure what is displayed (through panning and zooming, for example) or click on an area to reach another view. In the figure, the user has clicked on colony 105 in the lineage view and selected replicate 3, which links to the tabular display of data.

TOWARD PRODUCT-RELEASE CRITERIA

In our work, we hypothesized that product quality relates to a stem cell colony's purity, which we categorized as heterogeneous, homogeneous, or dark. We also hypothesized that quality relates to the overall distribution of colonies in each category and any

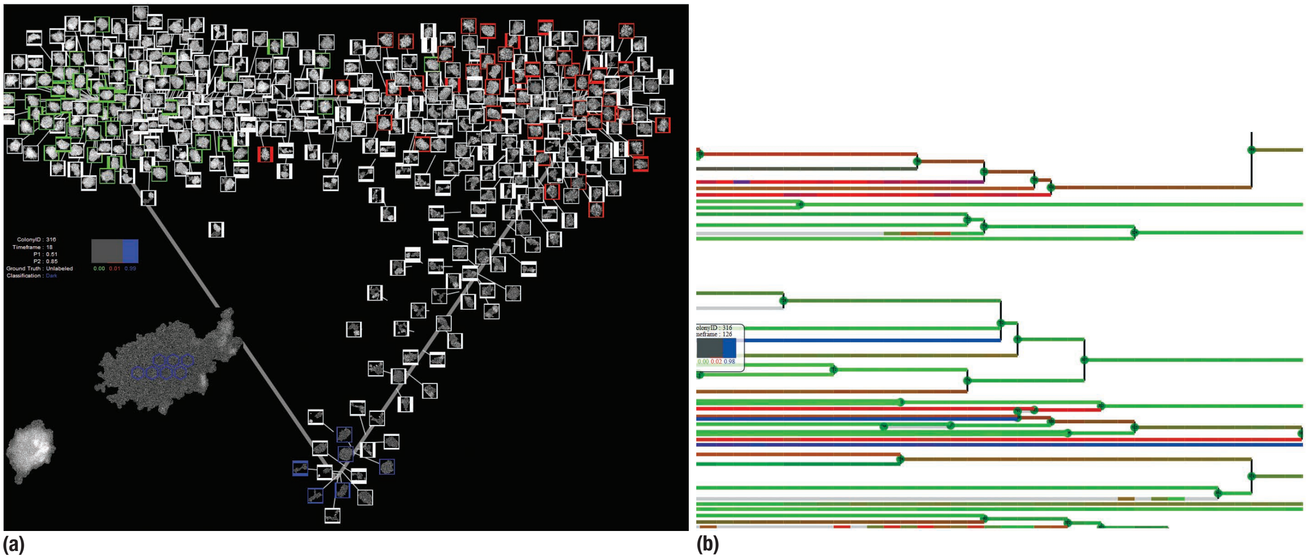


FIGURE 3. Interactive colony visualization. (a) Visualization of colony purity according to classification categories: homogeneous (green), heterogeneous (red), and dark (blue). (b) Visualization of how colony purity varies across time using the lineage tree and probabilities of each classification category per colony (inset). Visualizations such as these help scientists quickly assess stem cell colony characteristics.

temporal changes of category type over a colony's life.

To test our hypotheses, we collected 143 examples of colony categories provided by biologists using the data acquired from the three live stem cell preparations and then trained a logistic-regression classification model to assign categories automatically to all the colonies. The classification model's accuracy for the three datasets was 98, 87.5, and 100 percent.

Figure 3 shows interactive visualizations of the overall distribution and temporal changes of colony category type. The visualization coordinate system projects categories on a triangle and can be used to extract quantitative measurements that support visual quality assessments. The scientist can verify a colony classification by checking that its position on the triangle corresponds to its actual type. In Figure 3, colonies in the top left corner are likely to be bright homogeneous, colonies in the top right corner, heterogeneous (mixture of bright and dark intensities); and colonies on the bottom, dark (have a very low intensity similar to the background intensities).

CONFIGURATION AND COMPUTING TRADEOFFS

Designing a quantitative microscopy solution for characterizing stem cell products requires considering a number of tradeoffs during both offline computations and Web-based system configuration. Examples include using open source versus licensed software to run algorithms on a computing cluster and deciding which image bit depths (bits per pixel) are suitable for viewing in a browser versus use in scientific analysis.

Another set of tradeoffs concern collocating data and computational resources in desktop, cluster, and client–server environments. And to achieve interactivity in a client–server environment, scientists must balance computing image thumbnails on the fly versus precomputing them as well as interactive and non-interactive options for subsetting and image filtering.

Collocating data and resources offline

For offline image-processing computations, every algorithmic execution requires optimizations that are based on the hardware and network parameters. Part of optimizing is to decide whether to move the data to a computer cluster or to a powerful desktop. The tradeoff is labor (algorithmic redesign) and hardware cost versus higher computational speed with a cluster, which raises the question, "At what data size does a cluster become more efficient than a desktop?"

To answer that question, we experimented with existing open source and newly created Java-based algorithms running on a Hadoop cluster (850 Intel and AMD 64-bit nodes) and with custom Matlab-based algorithms running on a desktop with six physical cores (Intel Xeon CPU E5-2620 with a 2-GHz processor and 64-Gbyte RAM). The Hadoop Java-based algorithms computed image pyramids and performed flat-field correction, segmentation, reprojection, and image-feature extraction. The Matlab-based algorithms performed flat-field and background correction, stitching, segmentation, and colony tracking.

We placed the raw images for the three preparations, which we labeled R1, R2, and R3 in network attached storage (NAS) accessible through a 1-Gbit Ethernet LAN and in a NIST cluster connected through the same LAN. We observed the following transfer rates:

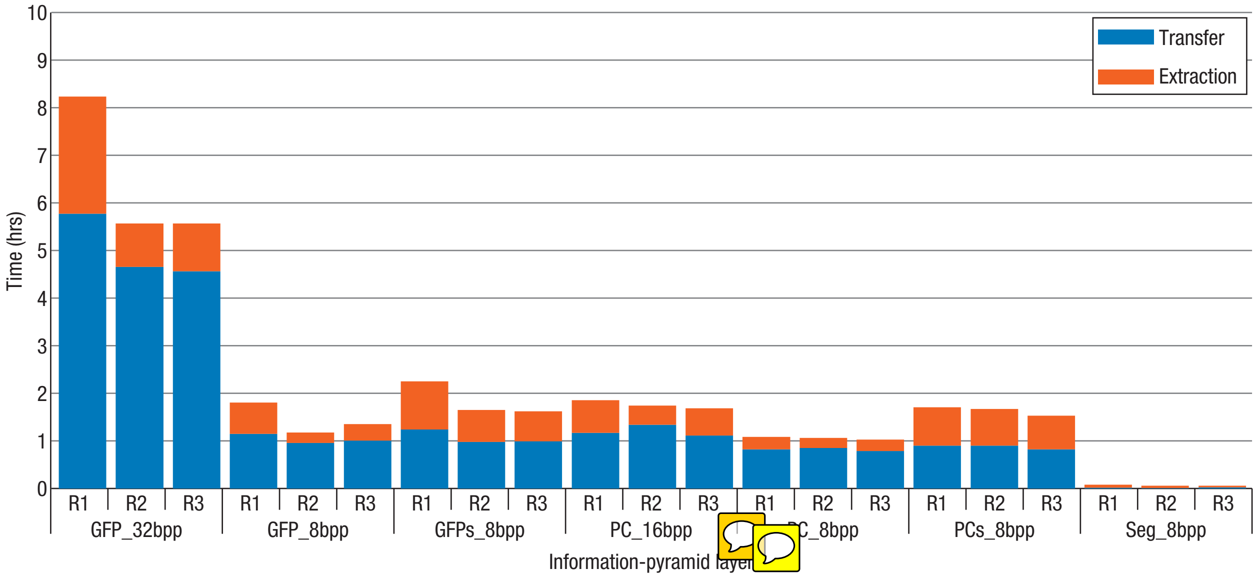


FIGURE 4. Times to transfer and extract pyramid sets for three biological preparations (R1, R2, and R3). Transfer was from a computer cluster to a storage array over a LAN. Each set is labeled by its image modality and bits per pixel (bpp). Image modalities shown are green fluorescent protein (GFP), phase contrast (PC), and segmentation (Seg). The "s" after GFP and PC denotes the reprojected side view.

- to NAS, 60 Mbytes per second (MBps);
- from NAS, 40 MBps;
- to NIST cluster, 5.3 MBps; and
- from NIST cluster, 2.7 MBps.

The ratios of NAS to NIST cluster transfer speeds are approximately 11.3 (60/5.3) and 14.8 (40/2.7), which are useful in a cost–benefit analysis of executing computations by attaching NAS with data versus transferring data to a computational resource.

For cluster computations that generate large output-file collections, as in multiresolution pyramid building, transferring data from a cluster to a Web server has a cost that depends on whether the data is compressed. We opted to package our pyramid files into a tar file to expedite file transfer. We started with 502,971 input files (192,085 in R1, 166,580 in R2, and 144,306 in R3) processed into 11 million output files located in a half million folders for pyramid representations of all layers. The time to move pyramid files for seven pyramid layers (including transfer and tar file extraction) from NIST cluster to NAS was R1, 19.61 hr; R2: 14.90 hr; and R3, 14.82 hr.

Figure 4 shows the time measurements per pyramid layer. The transfer times correspond to the following three steps: Use block sorting and Huffman compression to compress a set of pyramids representing one image layer into one tar file per pyramid, compress of one set of tar files into one big tar file per pyramid set, and transfer the compressed tar file.

The extraction time consists of two steps: extract a set of tar files from one big tar file and extract the image files from each tar file. The average ratio of transfer to extraction time was 2.5 (R1: 1.86, R2: 3.00, and R3: 2.63). These measurements were useful in comparing pyramid transfer with 8-bpp, 16-bpp, and 32-bpp images.

Web-based interactive computations

The main challenges in building a client–server system for large images or 3D volumes stem from limited storage, RAM, processing, and bandwidth. Optimal design decisions to achieve interactivity must be based on anticipated system use and available resources, yet system use is unpredictable in experiments to pursue scientific discovery. In our work, we assumed that system use would be primarily to

- disseminate data, including tasks such as browsing, reprojection rendering, and subsetting;
- extract basic statistical summaries, including comparisons of image features at colony and hexagon levels, and sorting;
- perform simple image filtering;
- take distance measurements at multiple length scales (resolutions); and
- annotate colonies with semantic labels.

To ensure that Web-based computation and measurement were sufficiently interactive, we chose to incur the higher cost of storing precomputed information on the server side instead of having more powerful computing resources for on-demand computation. We evaluated tradeoffs between on-demand thumbnail generation and storage costs of the precomputed thumbnail images. In addition, we explored the relative merits of interactive and noninteractive query-based execution for subsetting and image filtering.

Thumbnail images. We assessed two options for generating thumbnail images: retrieve a precomputed thumbnail image stored in the database or compute the requested thumbnail images on the fly using the colony images at full resolution. We represented the 31,312 cell colonies as thumbnails in three sizes 50 × 50, 75 × 75, and 100 × 100 pixels using approximately 1.12 Gbytes for both image modalities.

The 1.12 Gbytes of extra storage cost and 3.12 hrs to compute the thumbnails offline ended up being the better approach because thumbnail retrieval from the database was 10 to 15 times faster on average per single request than on-the-fly thumbnail creation.

Subsetting and image filtering. The subsetting function captures either the images rendered on the client side (JavaScript code) or the images retrieved on the server side (Java code). The two subsetting implementations represent tradeoffs between the degree of interactivity and capabilities. Image rendering in a browser enables interactive subsetting with only the rendering time overhead, but browser capabilities are limited to rendering 8 bpp and to saving data in client's RAM because writing directly to a client's hard drive is impossible. Moreover, a client typically has limited multithreaded execution.

Many of these disadvantages can be avoided on the server side by sacrificing some level of subsetting interactivity. In an experiment we collected runtimes for 3D subsetting with the Chrome 32 browser that show the execution time for server-side subsetting is 16.7 times faster than client-side subsetting (3 *s* versus 50 *s* per [*x*, *y*] cross section from a [*x*, *y*, *t*] volume.

Similarly, image-filtering operations can be executed on either the client or server sides. Client-side execution of these operations on retrieved images provides immediate feedback during parameter optimization and visual verification, but computational resources are limited. Consequently, operations must be restricted to an image subarea at the original resolution or to a large image at lower resolution. Our Web-based measurement system circumvents that restriction through pixel-level manipulations on top of the multiresolution pyramid representation, which it provides as a plugin to OpenSeadragon. Thus, a user can optimize parameters of image analyses interactively in a browser and then launch the analyses of the entire image on a more powerful computational resource than the client.

Our system for monitoring and characterizing stem cell colonies leverages cloud and cluster computing for offline computations and extends Deep Zoom for interactive viewing, subsetting, and measurements. Our work is primarily for scientists who operate a microscope, process microscopy images, and share computation in a cloud and client–server Web-based system, although the technology is applicable to any time-lapse imaging study of live cells. Other applications include the confocal laser scanning study of cell morphology and the coherent anti-Stokes Raman spectroscopy imaging study of fixed cells with any subset of dimensions.

Our work explored mainly offline and on-demand image analyses and focused on enabling Tbyte-sized image measurements. However, we also identified several pipeline steps that lack standard operating protocols or interoperability and that have unknown accuracy and uncertainty. Because microscope hardware and software varies widely, community-wide meetings and community consensus efforts are needed to unify microscope interfaces and to image data. Projects such as the µManager framework for open source microscopy software and the Open Microscopy Environment[10] are important for interoperability and to ensure that research results are reproducible. Similar efforts are needed to address the missing measurement infrastructure and eliminate the current dilemma—to measure or not to measure Tbyte images.

ACKNOWLEDGMENTS

The stem cell colony images were prepared and acquired courtesy of Kiran Bhadriraju at the NIST cell biology microscopy facility. We also thank Anne Plant, John Elliott, and Michael Halter from the Biosystems and Biomaterials Division at NIST, and Jing Gao and Mylene Simon from Software and Systems Division at NIST for their contributions.

REFERENCES

1. "Japanese Scientist Wants Own 'Breakthrough' Stem Cell Study Retracted," CBS News, 10 Mar. 2014; www.cbsnews.com/news/japanese-scientist-wants-own-breakthrough-stem-cell-study-retracted.
2. C.G. Begley and L.M. Ellis, "Drug Development: Raise Standards for Preclinical Cancer Research," *Nature*, vol. 483, 2012, pp. 531–533.
3. E.H. Williams, P. Carpentier, and T. Misteli, "The JCB DataViewer Scales

Up.," *J. Cell Biology*, vol. 198, no. 3, 2012, pp. 271–272.

4. A.D. Edelstein et al., "Advanced Methods of Microscope Control using μManager Software," *J. Biological Methods*, vol. 1, no. 2, 2014, art. 10; www.jbmethods.org/jbm/article/view/36/2.
5. T. White, *Hadoop: The Definitive Guide MapReduce for the Cloud*, 3rd ed., O'Reilly Media, 2012.
6. P. Bajcsy et al., "Terabyte-Sized Image Computations on Hadoop Cluster Platforms," *Proc. IEEE Int'l Conf. Big Data* (BigData 13), 2013, pp. 729–737.
7. S. Saalfeld et al., "The Collaborative Annotation Toolkit for Massive Amounts of Image Data (CATMAID)," *Max Planck Inst. Molecular Cell Biology and Genetics*, 2013; http://catmaid.readthedocs.io/en/stable/introduction.html.
8. T. Pietzsch et al., "BigDataViewer: Visualization and Processing for Large Image Data Sets," *Nature Methods,* vol. 12, no. 6, 2015, pp. 481–483.
9. J. Beyer et al., "Exploring the Connectome: Petascale Volume Visualization of Microscopy Data Streams.," *IEEE Computer Graphics and Applications*, vol. 33, no. 4, 2015, pp. 50–61.
10. D. Schiffmann et al., "Open Microscopy Environment and FindSpots: Integrating Image Informatics with Quantitative Multidimensional Image Analysis," *Biotechniques*, vol. 41, no. 2, 2006, pp. 199–208.

ABOUT THE AUTHORS

PETER BAJCSY is a computer scientist at NIST. His research interests include the automatic transfer of image content to knowledge, image processing, machine learning, and computer and machine vision. Bajcsy received a PhD in electrical and computer engineering from the University of Illinois at Urbana-Champaign. He is a Senior Member of IEEE. Contact him at peter.bajcsy@nist.gov.

ANTOINE VANDECREME is a computer scientist at NIST. His research interests include image processing and big data computations, distributed computing, Web services and Web application design. Vandecreme received an MS in computer science from Institut Supérieur d'Informatique de Modélisation et de leurs Applications (ISIMA). Contact him at ant.vand@gmail.com.

JULIEN AMELOT is a senior software engineer at GEICO. His research interests include image processing, machine learning, data mining, interactive visualization, and distributed computing for big data. While performing the research reported in this article, Amelot was a data scientist and software engineer at NIST. He received an MS in computer science from École Supérieure d'Informatique et Applications de Lorraine (ESIAL). Contact him at julien.amelot@gmail.com.

JOE CHALFOUN is a research scientist at NIST. His research interests include medical robotics as related to cell biology with an emphasis on dynamic behavior; microscope automation; segmentation; real-time tracking; and subcellular feature analysis, classification, and clustering. Chalfoun received a PhD in mechanical robotics engineering from the University of Versailles. Contact him at joe.chalfoun@nist.gov.

MICHAEL MAJURSKI is a computer scientist at NIST and an MS and PhD student in information systems at the University of Maryland, Baltimore. His research interests include image processing, microscope automation, computer vision, and big data computation. Majurski received a BS in music from Pennsylvania State University. Contact him at michael.majurski@nist.gov.

MARY BRADY is the manager of the Information Systems Group in NIST's Information Technology Laboratory. Her research interests include developing measurements, standards, and underlying technologies that foster innovation throughout the information life cycle—from collection and analysis to sharing and preservation. Brady received an MS in computer science from George Washington University. Contact her at mary.brady@nist.gov.

DISCLAIMER

Commercial products are identified in this document in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.