# Deep Zoom tool for advanced interactivity with high-resolution images

Paul Khouri-Saba, Antoine Vandecreme, Mary Brady, Kiran Bhadriraju, and Peter Bajcsy

Extending Deep Zoom technology allows the viewer to record a customized path through many image files, enabling further measurements.

Deep Zoom technology enables efficient transmission and viewing of images with large pixel counts.[1] Originally developed for 2D images by Seadragon Software, it was expanded by Microsoft Live Labs, and by Google to support Google Maps. Later, it was extended for 3D and other visualizations with open-source projects such as OpenSeaDragon.[2,3,4] Here we report on the extension of Deep Zoom to 2D+temporal data sets, retrieving image features, and recording fly-through image sequences (recorded simultaneously and viewed sequentially) from terabytes of image data.

Our work is motivated by analysis of live cell microscopy images that use about 241,920 image tiles (∼0.677 TB) per investigation. Each experiment is represented by $18 * 14$ spatial image tiles of two color channels (phase contrast and green fluorescent protein) acquired over five days every 15 minutes. With hundreds of thousands of image tiles, it is extremely difficult to inspect the current 2D+time images in a contiguous spatial and temporal context without preprocessing (calibration and stitching of multiple images), and without browser-based visualization using Deep Zoom. Other challenges include Deep Zoom pyramid-building (the process by which an image is created as a pyramid of tiles at different resolutions)[5] and storage issues (for every experiment there are about 6,091,378 pyramid files in 16,225 folders). Analyzing cell images requires the comparison of image intensity values across various channels, as well as additional layers of extracted information (intensity statistics over cell colonies, for example). A further challenge is extracting parts of a Deep Zoom rendering to examine interesting subsets for documenting, sharing, and to perform further scientific analyses.
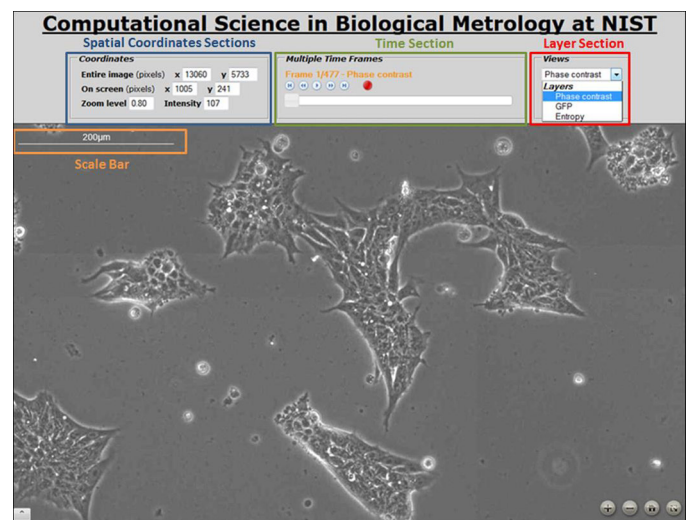
Figure 1. The main control panel for Deep Zoom image interactions, shown in a regular browser view.

We developed a visualization system called DeepZoomMovie that enables interactive capabilities using three sections of a browser toolbar (see Figure 1). The spatial coordinates section (displayed in the top left part of the toolbar) displays (X,Y) in pixels and shows the intensity of one pixel in a frame. It also displays the zoom level of the pyramid, defined as the ratio of the image's width to the viewport's width (min = 0.04, max = 2).
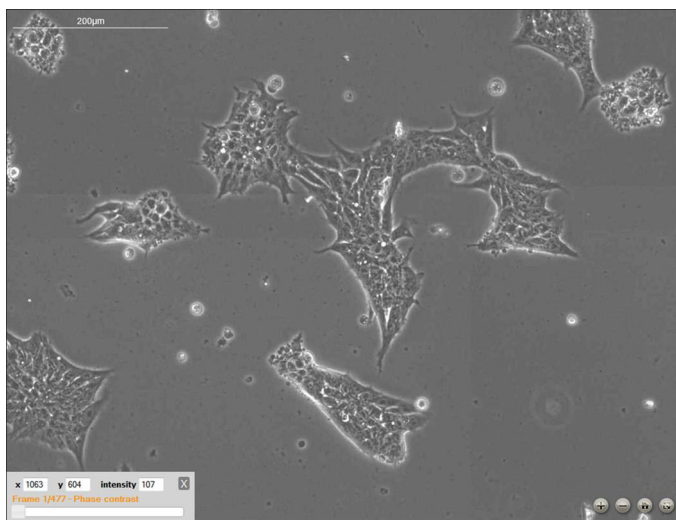
The time section (the middle-top part of the toolbar) displays the frame index over the interactive time slider and the video controls (play, pause, go to previous frame, go to next frame, record, go to first frame, and go to last frame). The save control is enabled in the recording mode, and saves not only the viewed images but also the image provenance information as a comma-separated value (CSV) file that contains the file name; layer name; frame index; zoom level; X, Y; width; and height of
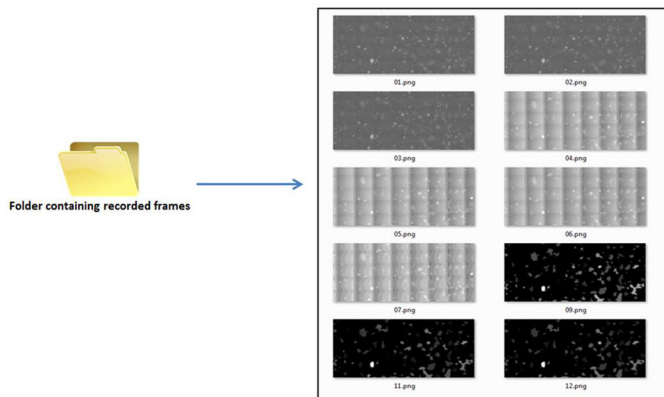
the recorded frame. The layer section (top right part of the toolbar) displays the drop-down menus for switching image layers and for changing the color of a scale bar.

All three sections are implemented on top of the SeaDragon JavaScript library. The DeepZoomMovie class is used in the same way as the SeaDragon Viewer class. To integrate it in a web page, we create a new DeepZoomMovie instance by specifying a parent container. To open a layer in the container, we call up the 'openLayer' method, with a JavaScript object notation item containing the characteristics of the layer, such as the path to the folder containing the DeepZoom images and the number of frames. The DeepZoomMovie class fires multiple events, so that a developer can interact with the DeepZoomMovie class and update other parts of the web page accordingly. Currently, the available events are: layerChanged, frameChange, frameChanged, animation and mouseStop. In our application, those events allow updating of the mouse position, the pixel intensity, and the zoom level.

To interact with the image we use the left (click and drag) and middle (zoom) mouse buttons via the full control toolbar (see Figure 1) in a regular browser mode, or a reduced tool bar in a full screen browser mode (see Figure 2). Additional features leverage JavaScript libraries such as the jQuery slider (time slider) and JSZip (packaging all recorded frames), as well as the JavaScript array storage, to manage the information being recorded (see Figure 3). Finally, we draw a scale bar on the rendered image to provide information on physical distances. We tested all functionalities in the latest versions of Firefox, Chrome, Safari, and Opera browsers.

Figure 3. A folder containing recorded frames. DeepZoomMovie images can be managed using JavaScript libraries and storage facilities.

We developed DeepZoomMovie to explore 2D+time large microscopy images, but it also enables big image data research in many other applications. We added the provenance information to support traceability of analytical results obtained over image subsets, and to enable statistical sampling of big image volumes with spatial and temporal contiguity. We have deployed the current capabilities at the National Institute of Standards and Technology on 1.8TB of test image data, where cell biologists are using them to explore the potential of stem cell colonies. In the future, we plan to extend the DeepZoomMovie code to enable distance measurements and annotations.

Disclaimer

Figure 2. Reduced control panel for image interactions in a full-page browser view.

Author Information

Paul Khouri-Saba, Antoine Vandecreme, Mary Brady, Kiran Bhadriraju, and Peter Bajcsy
National Institute of Standards & Technology
Gaithersburg, MD

Paul Khouri-Saba is a computer scientist working on a variety of software engineering topics, such as object-oriented programming, workflow execution, and imaging computations. His research interests include web development, mobile computing and visualization.

Antoine Vandecreme is a computer scientist working on image processing and big data computations. His research domains include distributed computing, web services, and web development.

Mary Brady is manager of the information systems group in the information technology laboratory. The group is focused on developing measurements, standards, and underlying technologies that foster innovation throughout the information life-cycle, from collection and analysis to sharing and preservation.

Kiran Bhadriraju is a bio-engineering research faculty member at the University of Maryland and studies cellular mechanotransduction and stem cell behavior. He has authored or co-authored 20 research publications.

Peter Bajcsy is a computer scientist working on automatic transfer of image content to knowledge. His scientific interests include image processing, machine learning, and computer and machine vision. He has co-authored more than 24 journal papers and eight books or book chapters.

References

1. http://en.wikipedia.org/wiki/Deep_Zoom Deep Zoom History. Accessed 5 May 2013.
2. http://openseadragon.github.com/#download OpenSeaDragon. Accessed 5 May 2013.
3. http://www.chronozoomproject.org/BehindTheScenes.htm Comparison of live cell imaging to cosmic chronology (ChronoZoom). Accessed 5 May 2013.
4. http://catmaid.org/ Comparison of Deep Zoom system to CATMAID The Collaborative Annotation Toolkit for Massive Amounts of Image Data. Accessed 5 May 2013.
5. R. Kooper and P. Bajcsy, *Multicore speedup for automated stitching of large images*, **SPIE Newsroom**, 1 March 2011. doi:10.1117/2.1201101.003451