

Biomedical Image Communication Platform

Masahiko Morita¹, Takehiro Tawara¹, Masaomi Nishimura¹, Shin Yoshizawa¹, Yukai Chou²,
Ippei Kuroki², Takashi Ijiri¹, Yuki Tsujimura¹, Ryuutaru Himeno² and Hideo Yokota¹

¹Image Processing Research Team, Center for Advanced Photonics

²Advanced Center for Computing and Communication
RIKEN, 2-1, Hirosawa, Wako, Saitama, 351-0198, Japan

Abstract—Recent progress in the field of biomedical imaging has led to the demand for large-scale three-dimensional (3D) image analysis in modern biology. Existing image-processing systems lack either cloud computing services or advanced 3D processing ability. This paper proposes a novel cloud-based communication platform for biomedical images. The system provides visualization, data management, and image-processing functions through a standard web browser. Therefore, the sharing of limited software and hardware resources is achieved, allowing for effective collaboration between researchers. We demonstrate the applicability and functionality of the system by examining typical case studies on biomedical images.

I. INTRODUCTION

Development of image-processing tools has become popular in biological and medical applications along with the rapid advances being made in biomedical imaging technology. In particular, tools that can handle large amounts of volumetric data (i.e., 3D images) are required in advanced applications such as surgery simulations based on Computed Tomography (CT) volumes and quantitative analysis of live-cell images in molecular cell biology. Many useful computational tools have been developed for biomedical image analysis, including ImageJ [1], CellProfiler [2], and Fiji [3]. However, unfortunately, most existing tools are stand-alone systems that are not equipped with a communication platform to allow collaboration among users, or the sharing of images, software, or CPU/GPU resources. Moreover, these conventional tools are either designed for a specific image-processing task or lack the 3D image processing functions required for advanced applications. Thus, biologists often buy or develop their own software and hardware for every research projects/groups. On the other hand, developing own image processing systems has been become difficult since expensive hardware, such as a GPU cluster and/or supercomputer, and its software are required for large-scale 3D image analysis.

In this paper, we propose a new cloud-based communication system for biomedical images which allows collaboration among users and enables effective and seamless management of images, software, and CPU/GPU resources. Our system consists of a set of computer servers such that a biomedical image database, a set of CPU/GPU servers, and a remote computing server, in order to provide the cloud computing services required for collaboration, management, and large-scale 3D analysis of biomedical images.

The key idea of our system is to employ standard web browsers (e.g. Internet Explorer, Google Chrome, and FireFox) as the user interface. We developed a novel cloud-based

interactive technique to provide user interactions and rendering results through the web browser, therefore our system is able to control the data management, visualization, and processing services through the web browser via inexpensive hardware devices such as tablet PCs, mobile phones, and cheap PCs without advanced GPUs, see Fig. 1. The image data are uploaded by using any standard FTP software to the biomedical image database. The user selects a CPU/GPU server through the biomedical image database with the image data. The CPU/GPU server is equipped with a set of image processing software (including our own [4], open sources, and commercials). The remote computing server controls user interactions in background processes on our system.

Our system also includes a unified way to manage biomedical images for collaborative research and an intuitive user interface for advanced processing of complex 3D biomedical images. We demonstrate our system by examining several case studies on biomedical images. The contributions and benefits of the paper are summarized as follows:

- A new communication platform for effective biomedical image analysis and management.
- An efficient way to share limited software and hardware resources.
- Supporting for a unified approach to collaborative research and 3D images.
- A novel cloud-based interactive technique by using a standard web browser for user interactions and rendering results.

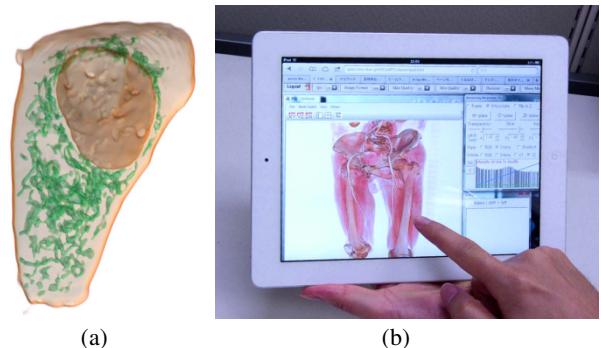


Fig. 1. (a): Intracellular volume rendering of the plasma- and nuclear-membranes and mitochondria by using our system. (b): Medical CT volume processing using our system via a tablet PC.

II. RELATED WORK

Image processing, including pattern recognition and computer vision, is one of the most intensively studied fields in computer science and therefore, a large number of software systems have been proposed over the past 60 years. Image analysis in biology has become popular in more recently compared with other scientific fields such as astronomical physics and geology. We briefly review the conventional biomedical image processing systems related to our system, see [5] and references therein for an excellent survey on general biomedical image processing tools.

ImageJ [1] has been a common tool for image analysis in biology, see [6] for its historical survey (although the Java programming language has only less than 16 years history in 2012). Cell Profiler [2] is a specialized tool for 2D segmentation and its quantitative analysis, and its extension [7] is able to handle and analyze multidimensional images by using machine learning techniques. Icy [8] and BioImageXD [9] also include multidimensional image processing ability; Icy optimizes its processing by multi-cores with the OpenCL implementation, and BioImageXD provides deconvolution and segmentation functions. All of them are very useful, but they are stand-alone software that do not provide any data shearing nor database functions. Any of these software can be used as an image processing engine on our CPU/GPU servers. In other words, our system is not a competitor of these stand-alone software but rather a collaborative engine in which they complement each other.

Cloud-based Systems: Many cloud-based systems have been proposed for some specific task such as CT reconstruction by using PC clusters [10] as well as for some particular technique such as transferring compressed images and videos [11], [12]. Our system is designed for general purposes in biomedical image visualization, processing, and analysis, see Fig. 2 for a representative example of 3D volume rendering with curved surface image visualization by our system.

In contrast to the specialized system, OMERO [13] and Bisque [14] are both web-based systems which mainly focus on data management for multidimensional images where they are accessed and used via a web browser (OMERO also has its own client) on Mac, Windows, and Linux. PSLID [15] is a database for 3D cell images which works on a web browser. Shen et al. [16] proposed a cloud-based classification tool for multidimensional biomedical images with a web browser interface. Unfortunately, the OMERO and Bisque only equip with simple image processing functions. Although the pattern recognition methods implemented in [16] are powerful such as a genetic algorithm and Support Vector Machine (SVM) and PSLID includes an image classification function, these conventional cloud-based systems [13], [14], [15], [16] lack 3D visualization and advanced 3D image processing abilities. In addition, it is difficult to incorporate conventional image processing tools [1], [7], [9] into these systems [13], [14], [15], [16].

Vazhenin [17] proposed a cloud-based system which allows the general populace to access medical-health information. The idea of his system's architecture is close to ours and his system includes 3D visualization. However, the purpose is very different such that our system aims at scientists and

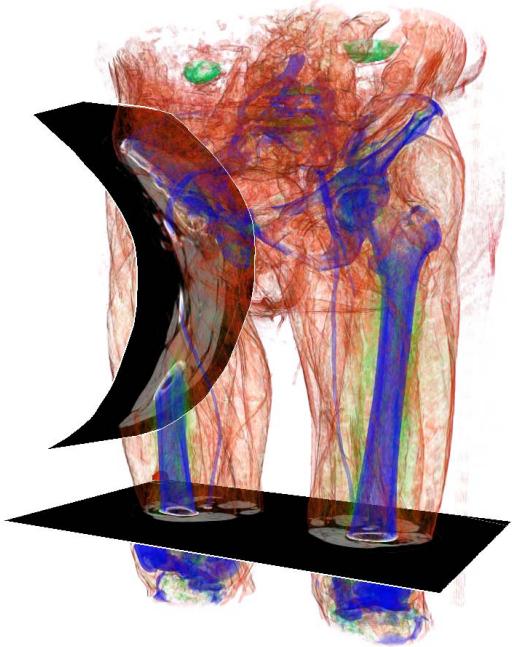


Fig. 2. Medical CT visualization by our system.

engineers who may not have knowledge about detailed image processing algorithms but they would like to apply advanced computations on their images. Therefore, our system is capable to incorporate image processing tools, besides the system of Vazhenin only provides visualization and is not equipped with advanced image processing tools.

III. PLATFORM OVERVIEW

A. System Services

Our system provides the 3D image data management, visualization, and processing services through a web browser. The main features of the services are as follows.

Data management: The system allows the user to store, search, and delete images with information which describes the experimental conditions for imaging and explanations of the data, e.g. description of what is observed in the image, microscopy parameters, image aspect-ratio, fluorescent staining methods, etc. The system also includes user accounts and their group management which controls permissions for the groups and users in order to support collaborative research and to manage software licences.

Visualization and processing: The user is able to employ any conventional image processing tool (e.g. [1], [7], [9]) that the administrator installs on the CPU/GPU servers. In the default setting, we install our software VCAT [4] as a core image processing engine. VCAT provides a set of plug-in image processing functions, its macro CAD interface, and visualization methods. Standard three-axial views and volume rendering with multi-dimensional transfer functions [18] are implemented in VCAT, see Fig. 1 (a) for an example of volume rendering for intracellular organelles and Fig. 3 for simple filtering examples.

Currently, the VCAT plug-ins include standard noise reduction filters (Gaussian, median, adaptive median [19], bilateral [20], and non-local means [21]), morphological operations [19], fast edge-aware filters [22], [23], [24], feature extraction methods (HoG [25], HLAC [26], Saliency [27], and SIFT [28]), SVM via the sequential minimal optimization [29], traditional (Otsu [30], graph cut [31], region growing [32], and mean shift [33]) and interactive [34], [35] segmentation methods, meshing segmented volumes (tetra- and hex-dominant tetra meshes, see Fig. 4), and other functions such as labeling, counting, and measuring segmented regions for quantitative analysis. VCAT employs a multi-page tiff format which is translated from standard biomedical image formats by using the Bio-formats library [36]. Of course, any other open-source/free/commercial software are able to be employed where their licences are managed by the system appropriately with user accounts.

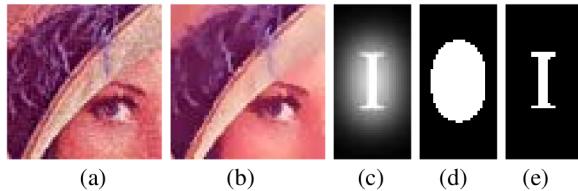


Fig. 3. Image processing examples by using VCAT. Denoising example (b) via the Domain Transformations (DT) [24] for a corrupted input image (a). Segmentation examples via the Otsu method [30] without (d) and with (e) Top-Hat transformation for an input image with a gradation of background intensity (c).

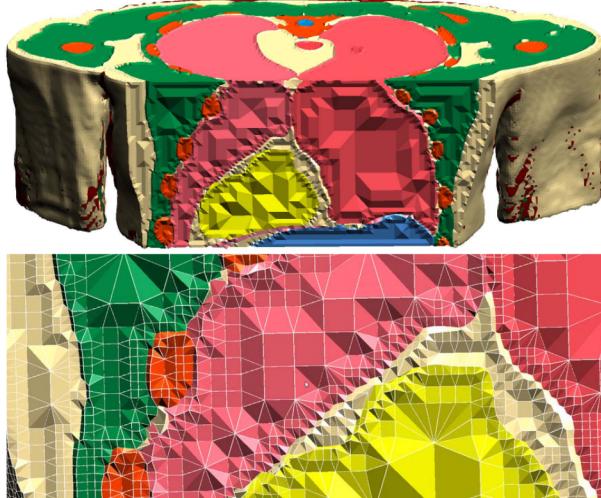


Fig. 4. Volumetric mesh generation for a multi-material biomedical volume. The hexa-dominant tetra mesh was generated on the multi-labeled segmented organ volume by VCAT.

B. System Architecture

The system consists of the following components to provide the above mentioned services.

- **Biomedical Image Database:** The database handles uploaded images with their additional information.
- **Remote Computing Server:** A server program handles user interactions from the web browser and rendering

results from the CPU/GPU Server.

- **CPU/GPU Servers:** High performance computers with advanced GPUs. The image processing engines (e.g. VCAT, ImageJ, etc.) are installed and a screen-capturing daemon process (capture server) runs on each server. The capture server sends the screen rendering results to the remote computing server.
- **Client:** Any standard PC including tablet PCs, mobile phones, and cheap PCs without advanced GPUs.

Figure 5 presents a flowchart describing how the image data and components interact.

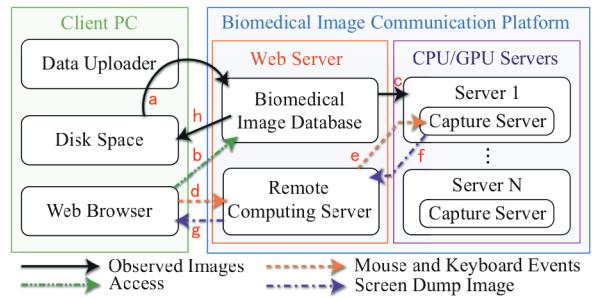


Fig. 5. Flowchart of data and interaction within our system architecture. (a,h): Image data are uploaded (downloaded) to (from) the biomedical image database via standard FTP software. (b): User logs in to the web server via the web browser to access the biomedical image database and selects a CPU/GPU server. (c): The selected CPU/GPU server downloads the images from the biomedical image database in a background process. (d,e): The remote computing server sends the user's mouse and keyboard events on the web browser to the CPU/GPU server. (f,g): The capture server (daemon process in the CPU/GPU server) automatically sends the desktop screen rendering results to the web browser through the remote computing server.

IV. CLOUD-BASED INTERACTIVE TECHNIQUE

Manipulating the image processing engines in the CPU/GPU server *interactively* from the client is crucial to cloud-based 3D image processing. It is especially important when the user attempts to adjust the volume rendering parameters (e.g. transfer functions and view information) and apply semi-manual segmentations [32], [31], [34], [35]. If the user is not able to manipulate the engines without delay and jumping of interactions and visualization results, it is extremely difficult to obtain the desired 3D image processing results (for quality evaluation, immediate visualization is important, even for time-consuming off-line filters). To resolve this issue, we developed a novel cloud-based interactive technique based on the following three schemes to improve the interactivity of the system.

Screen transfer by detecting difference: The capture server detects a change in the screen according to the differences, and then send the screen to the remote computing server if and only if it has changed. The capture server always holds an image of the previous screen. The sum of squared distances among uniformly sampled pixels between the previous and the current images (as specified in the configuration file) is computed for every frame. If the computed value is greater than a user-specified threshold, the current image is determined to have differences.

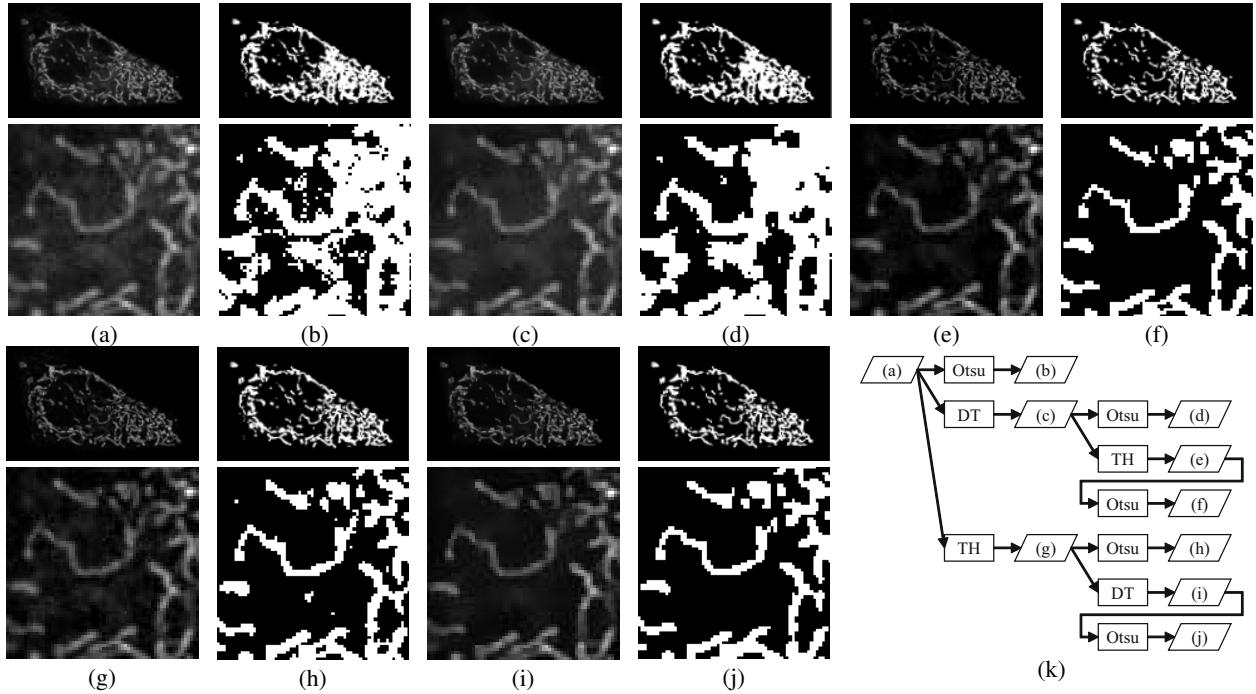


Fig. 6. Mitochondria segmentation results. All segmentations were obtained by the Otsu method. (a): Input image. (b): Segmented image of (a). (c): Denoised image of (a) via DT. (d): Segmented image of (c). (e): Top-Hat image of (c). (f): Segmented image of (e). (g): Top-Hat image of (a). (h): Segmented image of (g). (i): Denoised image of (g) via DT. (j): Segmented image of (i). (k): Processing flowchart with the resulting image IDs (a-j).

Spatially adaptive screen transfer: The desktop screen of the CPU/GPU server is subdivided, and each region is processed separately for sending the corresponding part of the screen to the remote computing server. The screen is divided into tiles, and the number of divisions can be changed dynamically via the menu displayed on the browser. The client's web browser periodically queries to the remote computing server as to whether the screen on the CPU/GPU server has changed. If it has changed, the CPU/GPU server sends the indices of the changed tiles (and their corresponding tile images) to the client's web browser. The client's web browser then updates only the corresponding tile images.

Quality adjusted screen transfer: The compression rate of the screen is dynamically changed according on user interactions. A low quality screen image is employed for high speed interactions and vice versa. This scheme also includes a buffer (queue) for the interaction events, so that stroke-based interactions, such as the drawing of curves, are able to be handled interactively. This interactive handling is important, for example, when editing transfer functions of volume rendering or sketch-based interfaces of [34], [35]. The image is compressed in JPEG format, and the image quality exponentially decreases between the user-specified maximum and minimum quality values during a mouse dragging operation. All tile images are rendered at the highest quality to guarantee the quality of the still image when the mouse button is released.

V. CASE STUDIES ON BIOMEDICAL IMAGES

In this section, we demonstrate our system by examining three case studies on biomedical images. The case studies

typify the use of our system: unsupervised segmentation and filtering, supervised machine learning, and interactive volume visualization and processing. All numerical experiments presented in this paper were executed on the following two workstations with VCAT. The biomedical image database and remote computing server were implemented on an Intel Xeon 3.3 GHz CPU workstation with 16 GB memory and Apache ver. 2.2.20 web server. The CPU/GPU server was an Intel Xeon 3.6 GHz CPU workstation with 64 GB memory and NVIDIA GeForce GTX 670. We employed an Intel Core2 Duo 1.83 GHz with 2 GB memory PC for the client, running Google Chrome as the web interface. The network was a 100 Mbps LAN.

A. Automatic Segmentation of Mitochondria

Figure 6 shows unsupervised image segmentations, typical of those used for intracellular images in the field of biology, of HeLa cells observed with a confocal laser scanning microscope and a fluorescent stain (Mito Tracker Deep Red 633). Domain Transformation (DT) [24] which is a state-of-art fast edge-preserving smoothing filter was employed to eliminate unessential noise, with the spatial and color range parameters set to 0.4 and 2000, respectively, to obtain the best results. The Top-Hat transform which consists of morphological operations [19] was applied to compensate for differences in intensity between the background and Mitochondria; a disk with radius 8 was used as the Top-Hat structural element. The segmentation was carried out by the famous Otsu's discriminant thresholding [30] which is the most frequently used unsupervised fully automatic segmentation method. As shown in Fig. 6, filtering with the Otsu method worked well for this case.

B. Organellar Classifications via SVM with HLAC

Supervised machine learning is a very powerful tool in computer vision and pattern recognition. We applied organellar classifications to three sets of images obtained by a confocal laser scanning microscopy with fluorescent staining (ER: Endoplasmic Reticulum (GFP), Golgi: Golgi Apparatus (RFP), and Mito.: Mitocondria (Mito Tracker Deep Red 633)). The image features were extracted by using HLAC [26] which approximates with higher-order local auto correlation on the image and is proven to be a very good image feature in pattern recognition. A fast implementation of a linear SVM [29] was employed for the classifier, in which a hyperplane in feature space was constructed such that the distance from the plane to the nearest feature was maximized. The maximum and minimum thresholds of the Lagrange multiplier were 100 and 0.001, respectively. Figure 7 shows representative training images for the SVM where 15 images were used for each class. Figure 8 illustrates the learning results of the ER vs. Golgi (a) and the ER vs. Mitocondria (b). As shown in Figs 9 and 10, the ER and Golgi were well recognized, while the system had some difficulty distinguishing between the ER and Mitocondria. In other words, the ER and Mitocondria were well correlated with each other. Further investigation is needed in order to understand the correlation in terms of biological evidences.

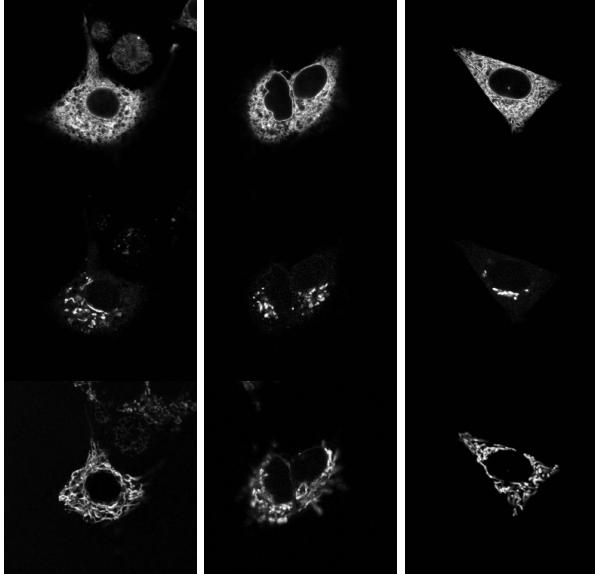


Fig. 7. Examples of training images for the SVM; top: ER, middle: Golgi, and bottom: Mitocondria.

C. Interactive Visualization and Segmentation of Medical CT

Data visualization is a key ingredient for image analysis. Our system provides volume rendering [18], orthogonal planer cross sections, and curved cross sections [34]. Figure 11 shows the volume rendering and curved cross section visualization of a CT image. Such high-quality volume visualizations usually require high-end, expensive computers. However, since our system allows the user to share the computer (i.e. the CPU/GPU server), it does not force he or she to have own one.

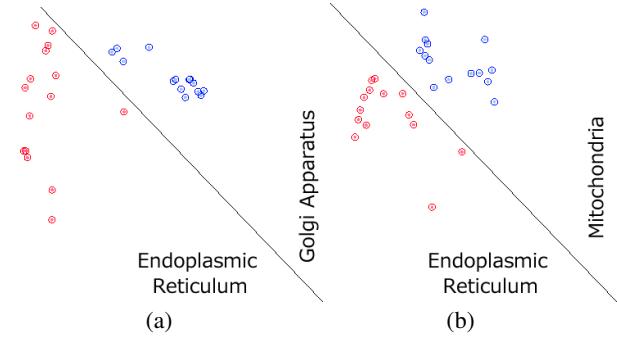


Fig. 8. Training results using the SVM with HLAC for the ER vs. Golgi (a) and the ER vs. Mitocondria (b). The line represents the SVM's hyperplane. The point coordinates are defined such that a distance to the hyperplane is equal to the normal direction of the hyperplane and the tangential direction is a distance from the projected point to a fixed point on the hyperplane.

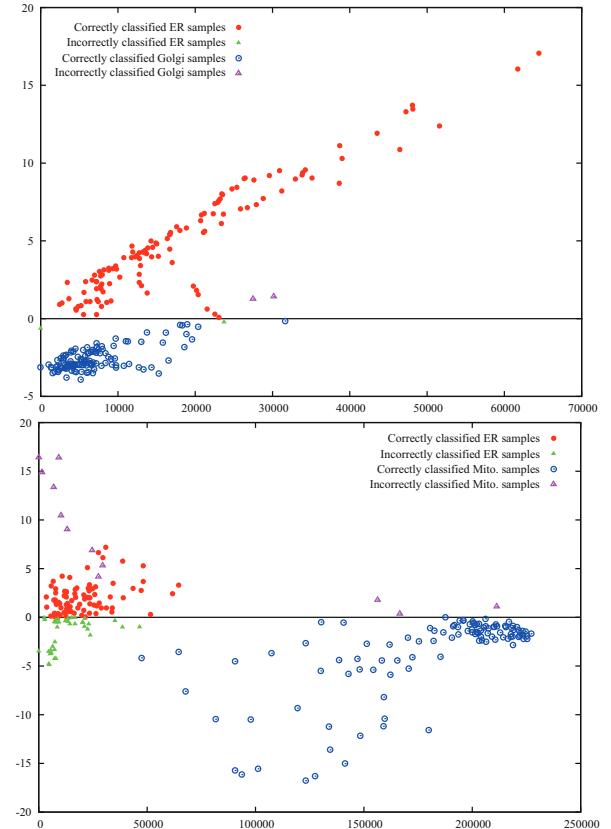


Fig. 9. Classification results for the ER vs. Golgi (top) and the ER vs. Mitocondria (bottom). The vertical and horizontal axes correspond to the distances described in the caption of Fig. 8.

	<i>N</i>	<i>M</i>	<i>A</i>		<i>N</i>	<i>M</i>	<i>A</i>
ER	122	120	98.36	ER	122	85	69.67
Golgi	120	118	98.33	Mito.	119	107	89.92

Fig. 10. Classification accuracy of Endoplasmic Reticulum (ER) with Golgi Apparatus (Golgi), and Mitocondria (Mito.) where *N* and *M* are the sample size and number of correctly classified images, respectively. The accuracy (*A*) is shown as a percentage (%) of the classifications.

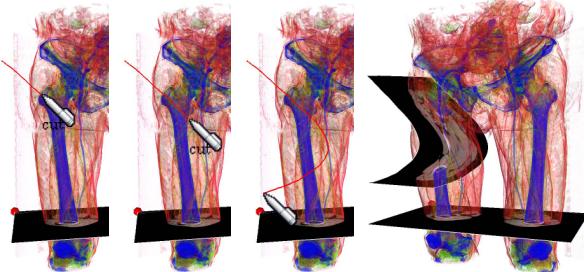


Fig. 11. Typical volume visualization using our system.

Fully automatic segmentation is difficult to apply to biomedical images since they often contain blurred boundaries and noise. Therefore, user input during the segmentation process is crucial. To help the user by providing subjective interpretation during the segmentation process, our system includes a set of interactive segmentation tools, such as seeded region growing [32], graph cut [31], and contour-based [35] segmentations. With the seeded region growing tool, the user places several seed points on the cross sections. The system gradually grows the foreground region by considering the local difference in voxel features (Fig. 12). With the graph-cut segmentation tool, the user places foreground/background seeds. Then the system computes the foreground voxels by minimizing an energy function. This method solves the energy minimization problem by using minimum cut of a flow network (Fig. 13). With the contour-based segmentation tool, the user specifies a set of contours in 3D space. The system generates a signed scalar field from the contours and obtains the segmentation boundary surface using the zero-level-set of the signed scalar field (Fig. 14). These interactive tools normally require high-end computing capacity. Our cloud system allows users to access these tools with inexpensive devices (e.g. tablet PCs).

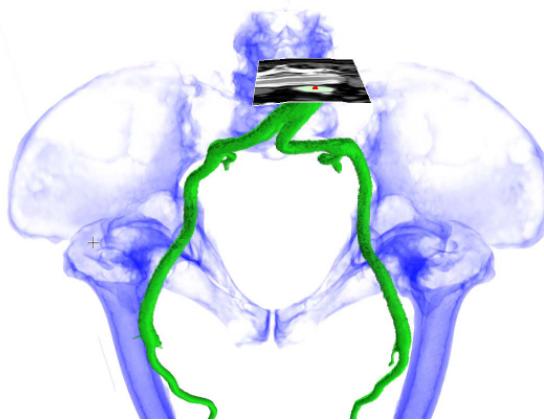


Fig. 12. The seeded region growing tool. The red mark is a seed. A segmented region is highlighted in green.

D. Performance Evaluation

We examined the performance of our cloud-based interactive technique described in Section IV. Figure 15 illustrates

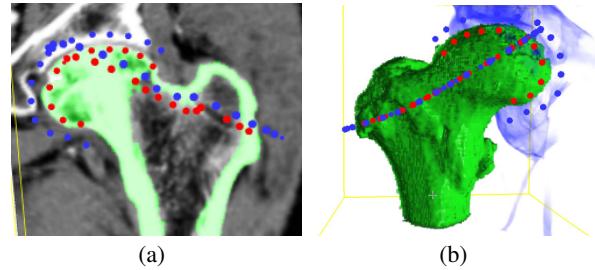


Fig. 13. The graph cut segmentation tool. The user specifies the foreground (red) and background (blue) seeds. The resulting segmented region is highlighted in green.

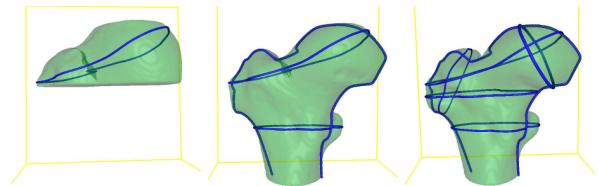


Fig. 14. The contour-based segmentation tool. The user specifies contours (blue), and the system computes a segmentation boundary (green surface).

the average response times and volume of downloaded data for the above three case studies (cases A, B, and C). The response times were measured through a web browser (Fig. 5). The display resolution was 1024×768 . For the spatially adaptive screen transfer scheme, a screen decomposition number of 6×6 was used. The maximum and minimum values for the quality-adjusted screen transfer scheme were 100% and 20%, respectively. The response time was not improved using the spatially adaptive screen transfer scheme. This may be due to a dilemma of some web browser specifications such that the response time depends on the number of loading images. Further investigation is needed to obtain the optimum spatial adaptive parameter. On the other hand, the screen transfer by detecting difference scheme worked well, as shown in Fig. 15 (D1-S1-Q0/1). These data indicate that our technique is useful and effective for the interactivity of a cloud-based system, which provides a rich variety of biomedical image-processing applications.

VI. CONCLUSION

We proposed a novel cloud-based communication system for biomedical images. The system is based on a set of computer servers that manage collaborative users, their images and software, and CPU/GPU resources effectively and seamlessly. Because our system is able to control data management, visualization, and processing services through a standard web browser on inexpensive hardware, users are able to collaborate and share limited software and hardware resources for biomedical image processing and analysis. The paper's technical contribution also includes the novel cloud-based interactive technique. We believe that our system is capable of significantly contributing to quantitative analysis in modern biology.

Future Work: We employed an intermediate upload server between the biomedical image database and the client FTP software in our current implementation. Future work will

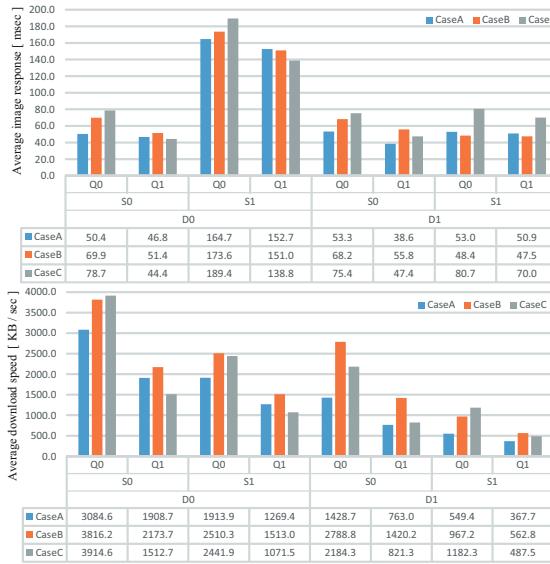


Fig. 15. Top: the average image response time (ms). Bottom: The average rate of data downloaded (KB/s). Here, D1/D0, S1/S0, and Q1/Q0 correspond to enable/disable of our screen transfer by detecting difference (D), spatially adaptive (S), and quality adjusted (Q) screen transfer schemes, respectively.

simplify this intermediate server and include more performance evaluation as well as evaluations via end-user studies.

Acknowledgements: We would like to thank Yutaka Ohtake for his meshing VCAT plug-in program and the anonymous reviewers of this paper for their helpful comments and suggestions. This work was supported in part by Strategic Programs for R&D (Presidents Discretionary Fund) of RIKEN and Grants-in-Aid for Scientific Research of Japan (20113007, 24700182, and 25700011).

REFERENCES

- [1] W. S. Rasband, "ImageJ: Image processing and analysis in Java," 1997-2013. [Online]. Available: <http://rsb.info.nih.gov/ij>
- [2] A. E. Carpenter et al., "CellProfiler: image analysis software for identifying and quantifying cell phenotypes," *Genome biology*, vol. 7, no. 10, p. R100, 2006.
- [3] J. Schindelin et al., "Fiji: an open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012.
- [4] N. Nishimura et al., "VCAT: Volume computer aided testing software ver.5," 2011-2013, RIKEN: Image Processing Research Team. [Online]. Available: <http://logistics.riken.jp/vcat/en>
- [5] K. W. Eliceiri et al., "Biological imaging software tools," *Nature Methods*, vol. 9, no. 7, pp. 697–710, 2012.
- [6] C. A. Schneider et al., "NIH image to ImageJ: 25 years of image analysis," *Nature Methods*, vol. 9, no. 7, pp. 671–675, 2012.
- [7] T. R. Jones et al., "CellProfiler Analyst: data exploration and analysis software for complex image-based screens," *BMC Bioinformatics*, vol. 9, no. 1, p. 482, 2008.
- [8] F. de Chaumont et al., "Icy: an open bioimage informatics platform for extended reproducible research," *Nature Methods*, vol. 9, no. 7, pp. 690–696, 2012.
- [9] P. Kankaanpaa et al., "BioImageXD: an open, general-purpose and high-throughput image-processing platform," *Nature Methods*, vol. 9, no. 7, pp. 683–689, 2012.
- [10] D. Thompson et al., "X-ray imaging software tools for HPC clusters and the cloud," in *Proc. of IEEE Int. Conf. on eScience*. IEEE-CS, 2012, pp. 1–7.
- [11] T. Lin and S. Wang, "Cloudlet-screen computing: a multi-core-based, cloud-computing-oriented, traditional-computing-compatible parallel computing paradigm for the masses," in *Proc. of IEEE int. conf. on Multimedia and Expo*. IEEE Press, 2009, pp. 1801–1804.
- [12] T.-Y. Wu et al., "Cloud-based image processing system with priority-based data distribution mechanism," *Computer Communications*, vol. 35, no. 15, pp. 1809–1818, 2012.
- [13] C. Allan et al., "OMERO: flexible, model-driven data management for experimental biology," *Nature Methods*, vol. 9, no. 3, pp. 245–253, 2012.
- [14] K. Kvilekval et al., "Bisque: a platform for bioimage analysis and management," *Bioinformatics*, vol. 26, no. 4, pp. 544–552, 2010.
- [15] R. F. Murphy, "Location proteomics: a systems approach to subcellular location," *Biochemical Society Transactions*, vol. 33, no. 3, pp. 535–538, 2005.
- [16] C.-P. Shen et al., "A multiclass classification tool using cloud computing architecture," in *Proc. of IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*. IEEE-CS, 2012, pp. 765–770.
- [17] D. Vazhenin, "Cloud-based web-service for health 2.0," in *Proc. of Joint Int. Conf. on Human-Centered Computer Environments*. ACM, 2012, pp. 240–243.
- [18] J. Kniss et al., "Multidimensional transfer functions for interactive volume rendering," *IEEE Trans. on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 270–285, 2002.
- [19] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007.
- [20] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. of ICCV*. IEEE-CS, 1998, pp. 839–846.
- [21] A. Buades and B. Coll, "A non-local algorithm for image denoising," in *Proc. of IEEE CVPR*, 2005, pp. 60–65.
- [22] F. Porikli, "Constant time O(1) bilateral filtering," in *Proc. of IEEE CVPR*, 2008, pp. 1–8.
- [23] S. Yoshizawa et al., "Fast gauss bilateral filtering," *Computer Graphics Forum*, vol. 29, no. 1, pp. 60–74, 2010.
- [24] E. Gastal and M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. on Graphics*, vol. 30, no. 4, pp. 69:1–69:12, 2011, proc. of ACM SIGGRAPH.
- [25] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of IEEE CVPR*, 2005, pp. 886–893.
- [26] N. Otsu and T. Kurita, "A new scheme for practical flexible and intelligent vision systems," in *Proc. of IAPR Workshop on Computer Vision*. IAPR, 1988, pp. 431–435.
- [27] L. Itti et al., "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [29] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in kernel methods*, B. B. Schölkopf et al., Ed. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.
- [30] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, pp. 23–27, 1975.
- [31] Y. Boykov et al., "Fast approximate energy minimization via graph cuts," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [32] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.
- [33] D. Comaniciu and P. Meer, "Mean Shift: A robust approach toward feature space analysis," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.
- [34] T. Ijiri and H. Yokota, "Contour-based interface for refining volume segmentation," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2153–2160, 2010.
- [35] T. Ijiri et al., "Bilateral Hermite radial basis functions for contour-based volume segmentation," *Computer Graphics Forum*, vol. 32, no. 2, pp. 123–132, 2013, proc. of EUROGRAPHICS.
- [36] U. W.-M. Biophotonics Research Lab., "Bio-formats library," 2013. [Online]. Available: <http://www.loci.wisc.edu/software/bio-formats>