Published in final edited form as:
Comput Biol Med. 2008 April ; 38(4): 490–500.

3D Medical Volume Reconstruction Using Web Services

Rob Kooper[*], **Andrew Shirk**[*], **Sang-Chul Lee**[*], **Amy Lin**[**], **Robert Folberg**[**], and **Peter Bajcsy**[*]

[*] *National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign (UIUC), 1205 W. Clark St., Urbana, IL 61801, {kooper,shirk,sclee,pbajcsy}@ncsa.uiuc.edu*

[**] *Department of Pathology, University of Illinois at Chicago (UIC), 1819 W. Polk Street,446 CMW, Chicago, IL 60612, {alin, rfolberg}@uic.edu*

Abstract

We address the problem of 3D medical volume reconstruction using web services. The use of proposed web services is motivated by the fact that the problem of 3D medical volume reconstruction requires significant computer resources and human expertise in medical and computer science areas. Web services are implemented as an additional layer to a dataflow framework called Data to Knowledge. In the collaboration between UIC and NCSA, pre-processed input images at NCSA are made accessible to medical collaborators for registration. Every time UIC medical collaborators inspected images and selected corresponding features for registration, the web service at NCSA is contacted and the registration processing query is executed using the Image to Knowledge library of registration methods. Co-registered frames are returned for verification by medical collaborators in a new window. In this paper, we present 3D volume reconstruction problem requirements and the architecture of the developed prototype system at http://isda.ncsa.uiuc.edu/MedVolume. We also explain the tradeoffs of our system design and provide experimental data to support our system implementation. The prototype system has been used for multiple 3D volume reconstructions of blood vessels and vasculogenic mimicry patterns in histological sections of uveal melanoma studied by fluorescent confocal laser scanning microscope.

Keywords

Web Services architecture; workflow; medical image analysis; test case

INTRODUCTION

In general, web services attempt to solve many of the same problems related to running distributed software applications as the more mature, previous generation, middleware technologies, e.g., CORBA and DCOM/COM+ [1][2]. When deciding how a distributed software application should be implemented, web services should be considered since they offer new advantages over the previous middleware technologies.

One of the key advantages of web services is real system interoperability. The real system interoperability is made possible by XML based standards, such as the Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL). SOAP and WSDL are simultaneously compatible with and independent of any particular programming language. Many distributed applications can tremendously benefit from using web services because there is a need to execute multiple processing algorithms on different operating systems and devices (potentially using Grid computing). Furthermore, web services provide benefits when software algorithms have been written using different programming languages and originated from multiple vendors (software houses). Thus, web services can meet the objectives for designing distributed software applications, such as interoperability of multi-programming language bits of software, provided by independent vendors and running on several operating systems and devices. These objectives can be achieved at a very low cost by using publicly available web service development frameworks.

Another major advantage of Web services is their orientation toward loosely coupled architectures. Web service clients (end user applications and other web services) can dynamically bind with web services to couple heterogeneous computer architectures at runtime. Thus, investigating real-life solutions using web services today will build the foundation of web service based cyber-infrastructure for the future [3], and enable creating complex composite workflow solutions. The objective of web service based workflows is to orchestrate runtime execution on multiple computer architectures by combining several algorithmic and workflow solutions.

The focus of this paper is on the web service based cyber-infrastructure system design to perform image registration, and specifically 3D medical volume reconstruction. The motivation for using web services is leveraged by the aforementioned advantages and described as follows. The technology of web services opens new opportunities for applications that have high demands on computational resources (storage and computation), require setting up sophisticated computer algorithms, and involve geographically distributed expertise from multiple disciplines [4]. We have found web services to be an especially attractive technology for academic research projects due to their open and interoperable nature that facilitates sharing, collaboration, and discovery. These characteristics of web services in a combination with the aforementioned advantages outweighed the risk of adopting current web service technologies [5].

Within the scope of cyber-infrastructure system design, our work aims at providing computational resources to end applications using web services, building tools for user interaction with images (e.g., image visualization, registration feature selection), and using web services for accessing sophisticated algorithms and for executing computationally intensive and memory demanding image processing queries. Our objective in this work is to provide either a set of developed software tools or the hardware resources at NCSA or both to scientific communities with the use of web services.

The remaining sections of this paper report on meeting the above objectives and on resolving the tradeoff issues related to building a prototype cyber-infrastructure system based on web services. To explain the prototype challenges, we provide a brief description of the D2K Web Service architecture, document the benefits of web services in the context of 3D medical volume reconstruction, formulate the problem of 3D volume reconstruction using I2K algorithms [6], and elaborate the system design issues when using web services.

PREVIOUS WORK

There exist commercial software packages that address the problems of image access and navigation with other than web service approaches. In the medical domain, these solutions are known as “Virtual Microscopes” and primarily owned by companies (like Bacus Laboratories, Inc. and Aperio Technologies). In the GIS domain, the solution for accessing all IKONOS aerial photos before 9/11 was developed by Microsoft (navigation capability without annotation or computation capability).

Among the most recent solutions using web services, we should mention a new suite of web service tools to facilitate multi-sensor investigations in Earth System Science that is sponsored by NASA [7], and the ArcWeb tools which are web services implemented for GIS data operations. The tools for NASA are developed based on a framework using grid workflows (known as SciFlo) [8]. Other workflow frameworks, like Kepler [9][10], have not been used for applications using web services. The ArcWeb services are proprietary, and focus primarily on accessing images with the size in the terabytes, and reducing data storage and maintenance costs.

In contrast to the previous work, the presented work is based on the data flow framework called D2K [11]. D2K is a visual programming environment and data flow execution engine developed at NCSA for data mining applications (prediction, discovery, and anomaly detection with data management and information visualization). The underlying 3D volume reconstruction algorithms came from a library of image analysis tools called I2K [6], also developed at NCSA. Our prototype system has been used in practice for 3D reconstruction of uveal melanoma tissues based on the NIH-funded collaboration between UIC and NCSA [12].

D2K WEB SERVICE ARCHITECTURE

Data to Knowledge (D2K) [11] is a flexible data mining and machine learning system that integrates analytical data mining methods for prediction, discovery, and deviation detection, with information visualization tools. It offers a visual programming environment that allows users to easily connect software modules together in a unique data flow environment. D2K supplies a set of software modules and application templates, along with a standard API (Application Programming Interface) for software module development. The software modules are reusable components that facilitate collaboration among developers. These modules and the entire D2K environment are written in Java for maximum flexibility and portability.

Figure 1 is a conceptual diagram of the many components that make up D2K. The three inner rings represent the core components, including the infrastructure. The outer ring represents the graphical user interface of the D2K Toolkit, the D2K Server as well as other applications that can be built using the core components.

- D2K Infrastructure is the processing engine that all other components utilize. It is responsible for controlling the scheduling and execution of knowledge discovery applications on the compute resources. The infrastructure also defines the D2K API.
- D2K Modules are software components that perform data preparation and transformation, implement discovery and learning algorithms, and assist in the interpretation and evaluation of results. A D2K Module is an independent unit that may take input and produce output.
- D2K Itineraries are essentially applications composed of D2K Modules connected together. Inputs from one module will feed into other modules. Itinerary complexity is limited only by the needs of your project.
- D2K Toolkit provides a graphical interface to create D2K Itineraries. These itineraries can be saved to use later in other applications, or can be executed in the D2K Toolkit. If modules require user interaction dialogs will appear where the user can input values, load files or examine (intermediate) results.
- D2K Server will execute itineraries that created by the D2K Toolkit without the graphical interface. The D2K Server will not allow user interaction but can run itineraries on high performance machines.

The D2K Web Service (WS) provides a WS-I Basic Profile 1.1 (http://www.wsi.org/Profiles/BasicProfile-1.1.html) compliant programming interface for executing D2K Itineraries on remote D2K Servers. Each D2K Web Service endpoint contains a library of registered itineraries available for execution by clients. For each itinerary, a pool of resources required for execution (Java classes, property files, etc.) is also stored. Associated with each itinerary definition is a list of D2K Servers that are eligible to process it. When service clients submit job requests, the D2K Web Service automatically handles the brokering of execution to an appropriate, and available, D2K Server. In return, the D2K Server requests from the D2K Web Service the resources it will need to process the job. While a job is processing, the D2K Web Service monitors its progress and persists any results that are produced. All communications between the D2K Web Service and D2K Servers occur over transmission control protocol (TCP) socket connections using D2K specific protocols.

CHOOSING A WEB SERVICE APPLICATION

Using web services makes creating applications easier since we can leverage from existing infrastructure and existing protocols. Web services also allow us to leverage from the expertise of geographically distributed people. However, not all applications will benefit from using web services. The ease of using existing protocols comes with a small price of extra bandwidth and additional processing time. In our case leveraging the D2K Toolkit requires our application to send data to the web service, which in turn will send the request for itinerary execution to a D2K server. The next sections provide experimental data obtained to quantify the extra overhead. These types of experimental data allow better understanding of the underlying technology characteristics and lead to optimal design of the prototype system for 3D Volume Reconstruction.

Timing

In this experiment we compared the time it takes to send a packet from an application to the D2K server using the D2K webservices and back with the time it takes to send the same packet to a special server using TCP and UDP. In the first case we had the application, D2K web service, D2K server, and both the TCP and UDP servers all running on the same machine. In the second case we had the application running on separate machine (located on separate subnet) from the machine that ran the D2K web service, D2K server and both TCP and UDP servers.

The code that was executed for the TCP and UDP servers is the same code that was used for a module that was encapsulated in an itinerary that ran on the D2K server. The only difference between the UDP, TCP and D2K web service (with D2K server) configurations is the code that is executed when sending the packet from the client to the server and back. Thus, the difference in timing is only due to the overhead associated with sending the packet between the server and client (and in the case of the D2K server the time it takes to spawn a separate process) but not the time it takes to process the packet since all the code responsible for responding to the packet is the same. The experiment will give the application developer a good understanding of the additional time that is required per request to the D2K server (using the D2K web service) as compared with the time it takes for specialized UPD or TCP servers.

Table 1 shows that using web services adds an overhead of 9.5 seconds to the application. If we would have an application sending a lot of packets that need to be processed in a timely manner then web services might not be the optimal solution. On the other side, if we would have an application generating a large amount of data that is transmitted in a single chunk, or data that require a large amount of computation, then web services might be the appropriate solution.

3D Volume Reconstruction Using Web Services

3D volume reconstruction is understood as the problem of mosaicking microscopy image tiles of one cross section, and aligning images of multiple cross sections to form a 3D volume of large data size. The resulting 3D volume can then be visualized to assist with diagnosis.

We developed a multi-step process for alignment that does not require the insertion of fiduciary markers into tissue that might distort the features of interest. The process is summarized in Figure 2 as follows:

1. Select a frame from within each 3D sub-volume to be used for alignment of the sub-volumes
2. Segment out closed or partially opened regions for matching
3. Compute features of segmented regions, e.g., centroids and areas
4. Find pairs of matching features
5. Select best sub-set of matched feature pairs
6. Compute alignment transformation parameters and transforming 3D sub-volumes into the final 3D volume
7. Refine alignment based on normalized correlation
8. Transform sub-volumes using optimal transformation
9. Enhance image intensities for visual inspection purposes

Although all processes can be automated as described in [13], some steps require expert-interactions for some clinical applications for verification purpose, such as feature matching and match selection steps.

Our proposed solution for a 3D volume reconstruction process includes (1) methods for selecting stack frame based on entropy and contrast criteria, (2) automated or semi-automated registration techniques for image alignment [14], and (3) an image enhancement technique to compensate the effects of photo-bleaching [15].

The described 3D medical volume reconstruction problem requires significant computer resources and human expertise in medical and computer science areas. UIC approached NCSA to help with the problem of the 3D Volume reconstruction. We view web services as the mechanism for establishing a collaborative environment between medical and computer science collaborators and combining their geographically distributed expertise. Web services also allow us to quickly prototype an application and send a URL to UIC to obtain fast user feedback. In our collaborative environment, the data of 3D cross sections are large in size, and the registration computation requirements are significant. Therefore, the above benefits of web services outweigh the timing overhead of web services.

PROBLEM DESCRIPTION

In a collaborative environment with medical and computer science collaborators, the goal is to reconstruct 3D medical volume from high resolution microscopy images of several cross sections. High resolution mosaic images of cross sections are formed from a large set of tiles, and then the mosaic images are aligned to construct a 3D volume. From a medical collaborator viewpoint, 3D volume reconstruction requires (a) setting up sophisticated 3D volume reconstruction algorithms and (b) computation and storage beyond the capability of a desktop computer, and therefore computer science expertise and resources. From a computer science collaborator viewpoint, 3D volume reconstruction requires selecting pairs of matching features for cross section alignment and hence medical expertise. Thus, there is need to develop a cyber-infrastructure environment where the computational resources and the expertise of remotely located medical and computer science collaborators can be integrated.

Application Scenario

We address the problem of 3D volume reconstruction in a collaborative environment by using web services. Our prototype system, shown in Figure 3, enables medical and computer science researchers to solve 3D volume reconstruction problems using web services.

The developed solution consists of the following workflow. First, a medical collaborator, e.g., from UIC, acquires images and sends data to his or her computer science collaborator. It is also possible that the medical collaborator uploads the data assuming a high bandwidth connection. Second, a computer program automatically mosaics image tiles, selects the most salient frame from each sub-volume, segments the selected frames and pre-computes centroids of all segments. The preprocessed images and centroid information are packaged for web access. Third, the medical collaborator will be notified about the URL designed for accessing and navigating the image data, as well as for selecting registration points and visualizing registration results.

The medical collaborator selects matching features for cross section alignment by using standard human computer interfaces (HCI), and our developed image navigation tools [16]. The points are saved at NCSA for use later when the final 3D volume is reconstructed. Once the medical collaborated has selected their points a query is send from UIC to NCSA to request registration computation. After the computation is completed, the results can be previewed by the medical collaborator. The preview shows how well the image alignment was done. If the medical collaborator is not satisfied with the image registration they can select more points or modify the already selected points. In the aforementioned workflow, all operations that require intensive computation are performed at NCSA (computer resource location), while all operations that require medical domain knowledge (image acquisition, registration point selection, and volume inspection) are performed at UIC (domain expertise location).

SYSTEM DESIGN USING WEB SERVICES

The overall architecture of the proposed system is illustrated in Figure 4. It is assumed that (a) all connections inside of NCSA are high speed, and (b) there might be additional high speed connections between the storage machines and the NCSA supercomputer. A user (a medical expert) will be able to interact with the system if they have a low bandwidth connection to the Internet. If the image size is larger than user’s RAM or screen size then the image access would be enabled by (a) creating a multi-resolution image pyramid, (b) tiling images at multiple resolutions, and (c) storing pyramid and tile information in a database. The current implementation assumes that the two image frames to be registered do not exceed available memory of the medical expert’s desktop.

The problem of image navigation is solved by (a) selecting and displaying image sub-areas at a chosen resolution, (b) panning through spatially large images using vertical and horizontal slider bars, and (c) selecting sub-set of bands to display to original data, pre-processed data (segmentation) and registered data. We should note that in our architecture, the user interface was implemented using Java applets. The reason for choosing Java applets comes from the fact that web services have been designed for web-based software interoperability but not for image visualization and data interaction purposes. The additional problem of registration feature selection is approached by providing tools for either pixel selection or region selection that is converted to a region centroid. The problem of intensive computation and extensive storage is tackled by (a) preparing a set of processing algorithms accessible by web services, and (b) using storage and computational resources at NCSA.

The proposed approach is based on the storage-computation paradigm, where a user is running only a “thin” client applet with small storage-computation resources and all storage demanding and computationally intensive operations are performed at NCSA. If a user would like to perform all operations at his desktop then he could install the entire system locally on his machine. We have not pursued the paradigm where data sets would be transferred to a local machine and computationally intensive processing of large size image data would be performed on a local machine.

Tradeoffs of System Design

When designing the prototype we tried to optimize the system so that an end user (the medical expert) would have a responsive system that would show the most up-to-date information with a limited amount of resources used. We considered the following four tradeoffs: image data transfer, image segmentation computation, image compression, image transformation.

Image data transfer—Image transfer can be executed as a transmission of the full size image vs. the use of image pyramids [17][18], in which the system would load the image as needed, for example, when a user zoomed in/out of the image and panned around the image. In general, medical images can be large in size. In our test case, we worked with images that are approximately 1300 × 2700 pixels, but could potentially be larger. Using image pyramids will reduce the initial transfer of data but will continuously download data of the image tiles.

Image segmentation computation—Segmentation of the input images can be performed on a server side or on a client side. We segment the input images in order to improve registration accuracy by replacing a pixel location with a more reliable region centroid location. The segmentation could be performed before a user selects a region (segmentation results have to be transferred) or after a user chooses a location to segment locally (computation has to occur on a client side). Segmentation on the client side is a CPU intensive task. Segmentation on the server will reduce the client CPU time but will increase the image size that needs to be transferred.

Image compression—Medical image data can be transferred compressed or uncompressed. Transferring compressed input images will result in smaller data transfer, but will require some client CPU time to decompress the images. Transferring uncompressed images will increase the data transfer requirements. In our prototype, compressed images led to a size reduction from 15.3Mb to 2.27Mb.

Image transformation—Computation of image transformation parameters and the transformed images can occur on a client side or on a server side. Once the expert has selected at least three points or regions in each image, we can calculate the image transformation parameters for an affine transformation and transform one image into the coordinate system of the other image. If the computation is performed on a client side, then there would be less network traffic but higher demands on the CPU usage on the client side. If it is performed on a server side, then there would be less CPU usage on a client side but more network traffic to transfer transformed image data.

In our application scenario, the objective was to create a prototype that would be very responsive to a medical expert, and it would not require significant computational resources on a client side. Thus, we tried to minimize the computer requirements for a medical expert and leverage the virtually unlimited resources of NCSA. Table 2 lists the tradeoffs we have considered during the system design. Our tradeoff considerations revolved around the following metrics: (1) CPU time on a client side, (2) RAM on a client side, (3) bandwidth for image transmission, and (4) an overhead associated with number of queries. According to Table 2, we chose the options that would limit the resources at the client (UIC) side. Our final choices were (1) to transfer full images, (2) to perform segmentation on a server side, (3) to use image compression before sending data, and (4) to compute image transformation parameters and transformed images on a server side. The images used in our prototype were small enough in size to fit into RAM memory of a standard desktop computer, and hence we decided not to use the image pyramids.

Registration Decision Support via Web Services

To achieve optimal 3D medical volume reconstruction during web-service, we considered multiple registration variables and their impact on (a) anticipated accuracy of aligned images, (b) uncertainty of obtained results, (c) repeatability of alignment, and (d) computational requirements. The registration decisions could include (1) image size used for registration, (2) transformation model, (3) invariant registration feature (intensity or morphology), (4) automation level, (5) evaluations of registration results (multiple metrics and methods for establishing ground truth), and (6) assessment of resources (geographically local or distributed computational resources and human expertise) as illustrated in Figure 5.

The proposed decision support system provides data-driven mechanisms for evaluating the tradeoffs between accuracy of 3D volume reconstructions and registration variables. First, we present links between registration decisions and 3D reconstruction results in terms of accuracy, uncertainty, consistency and computational complexity characteristics. Second, we have built software tools that enable geographically distributed researchers to optimize their data-driven registration decisions by using web services and high performance computing (HPC) resources. The support developed for registration decisions about 3D volume reconstruction is available to the general community with the access to the NCSA HPC resources.

We describe data-driven optimization approaches to registration decisions including image size selection, rigid or affine transformation model, intensity or morphological invariant feature selection, and manual (pixel-based) or semi-automated (centroid-based) automation level. The reason for using data-driven approaches lies in the large variability of objects of interest, specimen preparation, imaging modality, specific instrumentation characteristics, and so on, that is extremely difficult to model analytically with any generality whatsoever. We frame the problem first by presenting the links between 3D reconstruction accuracy and registration variables. Then, we lead to the data-driven methods for optimal choice of registration variables. The experimental results illustrate how the developed web-enabled software would be used by researchers to make the most optimal data-specific registration decisions.

Prototype Solution for 3D Volume Reconstruction

The prototype solution is available at http://isda.ncsa.uiuc.edu/MedVolume/. We have tested the prototype with the Microsoft Internet Explorer 6.0 using the Sun Java Runtime Environment (JRE) 1.4.2 browser plugin. The left lower corner of the browser conveys messages about the execution steps, e.g., Applet ncsa.uic.RegistrationApplet started. A user can select image frames to register using the drop down menu with image names. Selected images are compressed at NCSA site, transferred to a client site, decompressed and displayed in the image panels (see Figure 6).

After selecting at least three pairs of matching points/segments, the compute button is enabled and the D2K Web Service can be contacted. If an image panel is in the point selection mode then the left mouse click will define the pixel location to be used for registration. If an image panel is in the segment selection mode then the left mouse click will be replaced with the centroid location of the segment that contains the mouse click location. Selection of registration points using the centroid feature approach is illustrated in Figure 7.

After launching web services by pressing the button “Compute”, a new Java window labeled “Executing Job” will appear containing information about the compute job while the affine transformation parameters are computed. When the job is completed, the button “View Results” becomes active and the results can be visualized on the client side as it shown in Figure 8. The visualization will contain a seven band image that was sub-sampled in order to decrease the file size. The seven band image contains the original left image bands, the transformed right image bands, and one black band for visualization purposes.

If the registration accuracy is satisfactory to the medical collaborators then they record the job ID number displayed in the top section of the screen. This ID number allows computer scientists at NCSA to retrieve the points associated with the correct registration session and complete the 3D volume reconstruction. Based on the set of points selected by medical experts and saved at NCSA, multiple confocal laser scanning microscope sub-volume are transformed into a reference coordinate system at NCSA and the 3D volume reconstruction results are posted for downloading at an ftp site or a web site.

EXPERIMENTAL EVALUATION

We performed two user studies to see how well our solution worked. NCSA received from UIC two datasets. Table 3 gives some information about the size of the datasets. The data delivered to NCSA was just the tiles for each dataset. Some preprocessing was needed before the datasets could be used in the application described previously.

The first preprocessing step done was to mosaic the tiles into frames. Figure 9 shows a single frame selected from the each dataset after the mosaicking. The fully automated mosaicking preprocessing step took one hour for dataset 1 and four hours for dataset 2. When scanning the sample, each tile was scanned at multiple depths resulting in a subvolume. The next step is to combine the frames into the subvolumes again. Next the best frame of each subvolume is selected and segmented [16] (4 hours for each dataset). The information about the segments and the centroid for each segment is packaged with the frame for use in the web service application (1 hour for each dataset).

The URL of the web service application is send back to UIC for the medical expert to do the alignment of frames. The medical expert will look at the frames and find corresponding points in each successive frame and view the results. This process took 140 minutes for dataset 2 and 129 minutes for dataset 1. Even though dataset 1 contains more subvolumes, and hence more frames that need to be aligned, this dataset was finished faster than dataset 2. This could be explained by the fact that (a) the dataset 1 contains more visually salient features than the dataset 2 and (b) dataset 2 was the first dataset processed by the medical expert and part of the extra time could be attributed to the learning process of the new tool.

After selecting the registration points an email message was send to NCSA with the job numbers of the best alignments. These jobs are then used to extract the points selected by the medical expert and used to create the final 3D volume. Dataset 1 took 80 minutes to create the final 3D volume of 2746 × 1170 × 208 with 2 colors for each. Dataset 2 took 60 minutes to create a 3D volume of 2300 × 2300 × 130 with again 2 colors per channel. Figure 10 show what the final 3D volume would look like for Dataset 1.

SUMMARY

In this paper, we presented a prototype solution for 3D medical volume reconstruction that was used in practice by UIC and NCSA collaborators. We overviewed the 3D volume reconstruction problem requirements, the architecture of the developed prototype system using web services and the tradeoffs of our system design.

In a summary, the web services based approach provides two major advantages. First, a user will be able to perform computationally intensive image operations (a) with large size image data and (b) with sophisticated 3D volume reconstruction analysis methods. Furthermore, a user will not have to invest into (a) computational and storage resources and (b) development of complicated analysis software. Second, the currently advertised interoperability feature of web services will enable us in the future (a) to customize system front end (graphical user interfaces (GUI) for the user entry point) without changing system back end (complex algorithms that perform desired computations), (b) to upgrade algorithms and fix software bugs without any involvement of a user, and (c) to integrate distributed web services that will be available on the Internet.

We also plan to add web services-based software tools for guiding medical researchers during 3D volume reconstruction. There are several registration decisions that have to be understood and optimized, for example, (1) image spatial size (image sub-area or entire image), (2) transformation model (e.g., rigid, affine or elastic), (3) invariant feature (intensity, morphology or sequential combination of the two), (4) automation level (manual, semi-automated, or fully-automated) and (5) evaluation criteria for registration results (metrics and methods for establishing ground truth needed for evaluations). Our work in progress involves adding tools to the URL: http://isda.ncsa.uiuc.edu/MedVolume/ for transformation model selection and sub-area selection. These tools use web services to perform intensive computation at NCSA, let medical experts to interact with images, and provide input for registration decisions.

Acknowledgements

This material is based upon work partially supported by the National Laboratory for Advanced Data Research (NLADR) and the National Institute of Health under Grant No. R01 EY10457. The ongoing research is collaboration between the Department of Pathology, College of Medicine, University of Illinois at Chicago (UIC) and the Image Spatial Data Analysis Group, National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana-Champaign (UIUC). We acknowledge NCSA/UIUC support of this work.

References

1. F. Bolton, Pure CORBA, Sams Publication (2001).
2. J. Pritchard, COM and CORBA(R) Side by Side: Architectures, Strategies, and Implementations, Addison-Wesley Professional (1999).
3. D.E. Atkins, K.K. Droegemeier, S.I. Feldman, H. Garcia-Molina, P. Messina, D.G. Messerschmitt, J.P. Ostriker, M.H. Wright, Revolutionizing Science and Engineering through Cyberinfrastructure: Report of the National Science Foundation, Blue ribbon advisory panel on Cyberinfrastructure, April 19, (2002).
4. N. Leavitt, Are Web Services Finally ready to Deliver?, p. 14–18, IEEE Computer, Computer Society, (2004).
5. Parastatidis, S.; Webber, J. Assessing the Risk and Value of Adopting Emerging and Unstable Web Services Specifications, Services Computing. IEEE International Conference on (SCC'04); 2004.
6. Bajcsy P, Kooper R, Clutter D, Lee S-C, Ferak P. Image to Knowledge user manual, ALG-2005-005. 2005
7. Yunck, T.; Wilson, B.; Braverman, A.; Dobinson, E.; Fetzer, E. GENESIS: The General Earth Science Investigation Suite. The fourth annual NASA's Earth Technology Conference; 2004.
8. Wilson, BD. GENESIS SciFlo: Enabling Multi-Instrument Atmospheric Science Using Grid Workflows. poster SF31A-0716 0800h at American Geophysical Union (AGU) Fall Meeting Special Focus: Advances in Data Acquisition, Management, Analysis and Display; 2003. p. 13-17.
9. Altintas, I.; Berkley, C.; Jaeger, E.; Jones, M.; Ludäscher, B.; Mock, S. System demonstration, Kepler: An Extensible System for Design and Execution of Scientific Workflows. 16th Intl. Conf. on Scientific and Statistical Database Management (SSDBM'04); 21–23 June 2004; Santorini Island, Greece. 2004.

10. Ludäscher B, Altintas I, Berkley C, Higgins D, Jaeger-Frank E, Jones M, Lee E, Tao J, Zhao Y. Scientific Workflow Management and the Kepler System, Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows. 2005

11. Welge, M.; Hsu, WH.; Auvil, LS.; Bushell, C.; Martirano, J.; Redman, TM.; Tcheng, D. Data to Knowledge (D2K): A Rapid Application Development Environment for Knowledge Discovery in Database. Technical Report, National Center for Supercomputing Applications; University of Illinois at Urbana-Champaign, Champaign. 1999.

12. Kooper, R.; Shirk, A.; Lee, S-C.; Lin, A.; Folberg, R.; Bajcsy, P. 3D Volume Reconstruction Using Web Services. The 3rd IEEE International Conference on Web Services (ICWS 2005); July 12–15, 2005; Orlando, Florida. 2005. p. 709-716.

13. Bajcsy P, Lee S-C, Lin A, Folberg R. 3D Volume Reconstruction of Extracellular Matrix Proteins in Uveal Melanoma from Fluorescent Confocal Laser Scanning Microscope Images. Journal of Microscopy, Blackwell Synergy 2006;221(1):30–45.

14. Lee SC, Bajcsy P, Lin A, Folberg R. Accuracy Evaluation for Region Centroid-Based Registration of Fluorescent CLSM Imagery. EURASIP JASP, Performance Evaluation in Image Processing, Hindawi publishing 2006;2006:1–11.article ID 82480

15. Lee SC, Bajcsy P. Intensity Correction of Fluorescent Confocal Laser Scanning Microscope Images by Mean-Weight Filtering. Journal of Microscopy, Blackwell Synergy 2006;221(2):122–136.

16. Lee, S-C.; Bajcsy, P. Feature based Registration of Fluorescent LSCM Imagery, SPIE International Symposium in Medical Imaging. Proceedings of SPIE Conference on Medical Imaging; 12–17 February 2005; San Diego, CA. 2005.

17. Rosenfeld, A. Multiresolution Image Processing and Analysis. Springer-Verlag; New York: 1984.

18. Cantoni, V.; Levialdi, S. Pyramidal Systems for Computer Vision., NATO ASI Series, Series F: Computer and Systems Sciences, 25. Springer-Verlag; New York: 1986.

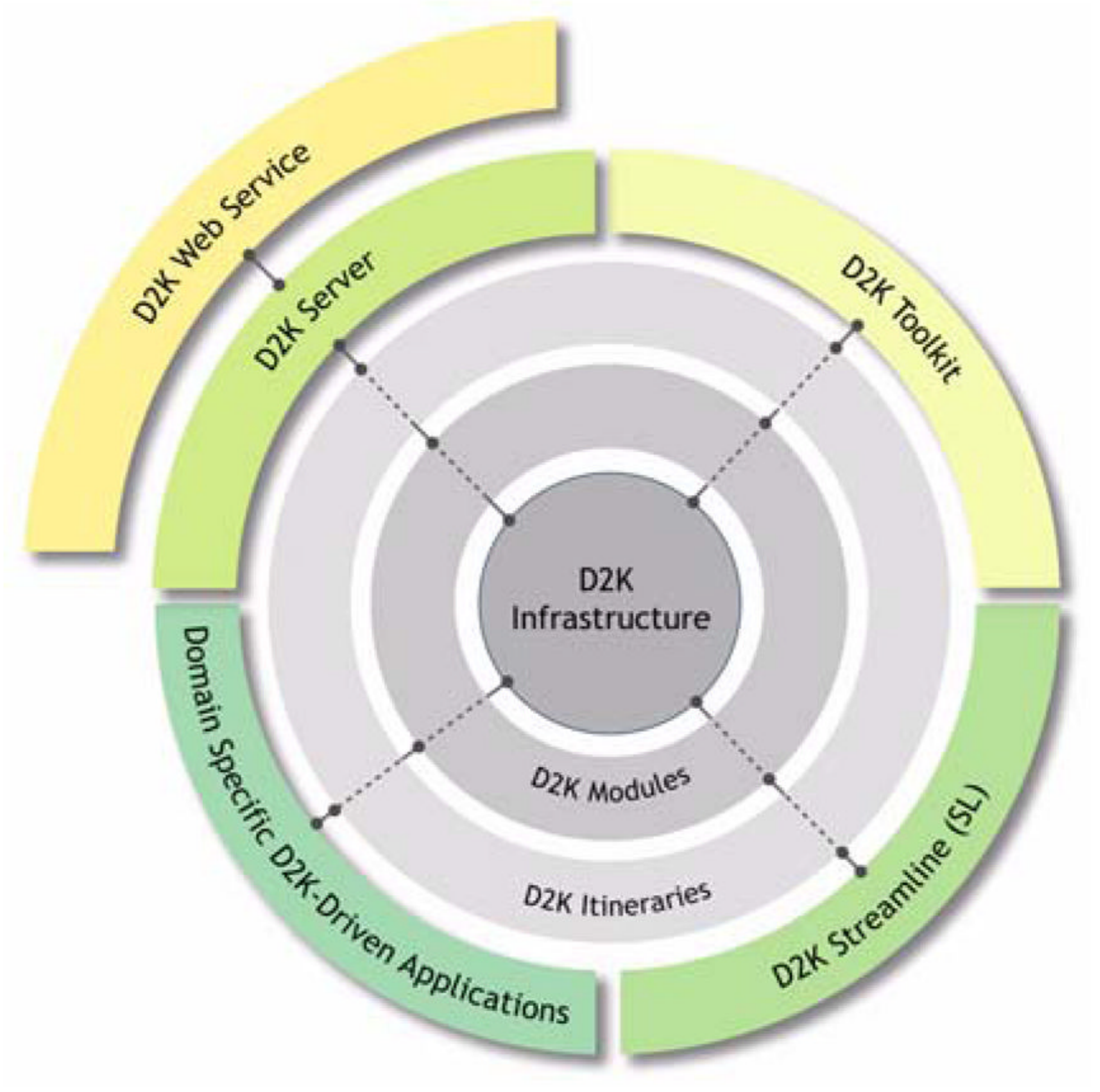


Figure 1.
D2K component architecture

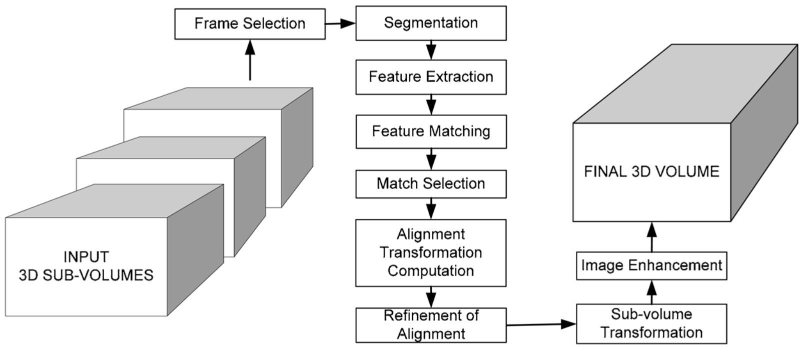


Figure 2.
An overview of 3D volume reconstruction steps from input fluorescent confocal laser scanning microscope 3D sub-volumes. The processing steps start with a set of sub-volumes and end with one registered 3D volume.

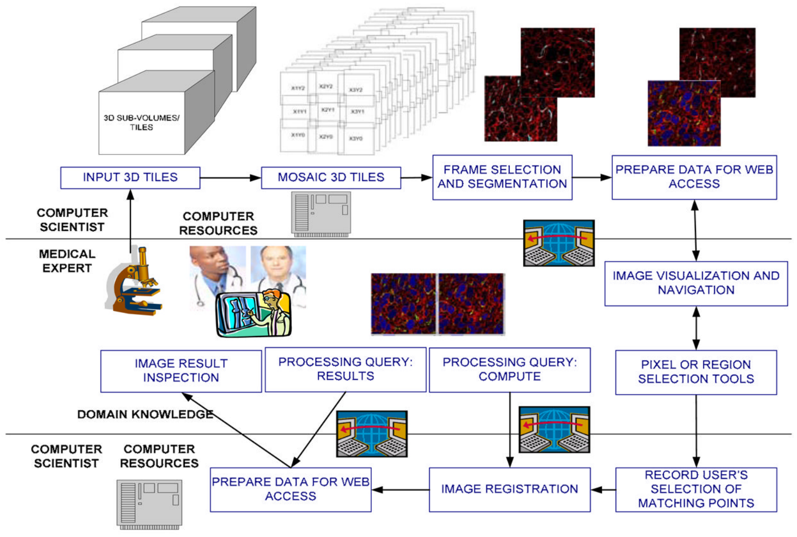

3D SUB-VOLUMES/ TILES
INPUT 3D TILES
MOSAIC 3D TILES
FRAME SELECTION AND SEGMENTATION
PREPARE DATA FOR WEB ACCESS
COMPUTER SCIENTIST
COMPUTER RESOURCES
MEDICAL EXPERT
IMAGE VISUALIZATION AND NAVIGATION
IMAGE RESULT INSPECTION
PROCESSING QUERY: RESULTS
PROCESSING QUERY: COMPUTE
PIXEL OR REGION SELECTION TOOLS
DOMAIN KNOWLEDGE
COMPUTER SCIENTIST
COMPUTER RESOURCES
PREPARE DATA FOR WEB ACCESS
IMAGE REGISTRATION
RECORD USER'S SELECTION OF MATCHING POINTS

Figure 3.

An overview of the current application scenario.

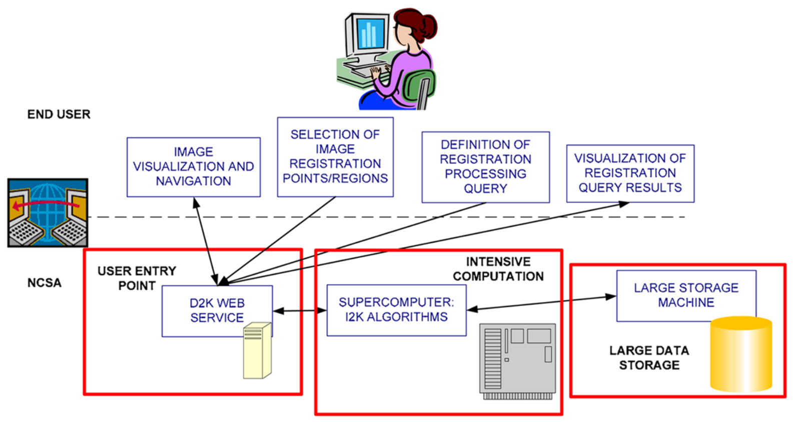

END USER
IMAGE VISUALIZATION AND NAVIGATION
SELECTION OF IMAGE REGISTRATION POINTS/REGIONS
DEFINITION OF REGISTRATION PROCESSING QUERY
VISUALIZATION OF REGISTRATION QUERY RESULTS
NCSA
USER ENTRY POINT
D2K WEB SERVICE
INTENSIVE COMPUTATION
SUPERCOMPUTER: I2K ALGORITHMS
LARGE STORAGE MACHINE
LARGE DATA STORAGE

Figure 4.
The overall architecture of the proposed system.

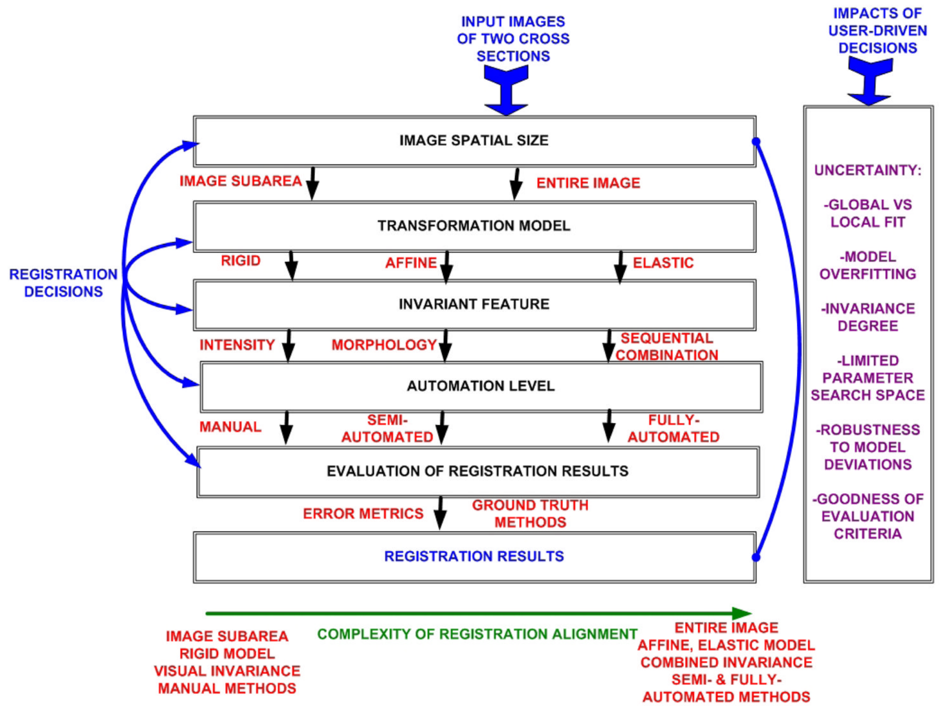

INPUT IMAGES OF TWO CROSS SECTIONS
IMPACTS OF USER-DRIVEN DECISIONS
IMAGE SPATIAL SIZE
IMAGE SUBAREA
ENTIRE IMAGE
TRANSFORMATION MODEL
RIGID
AFFINE
ELASTIC
INVARIANT FEATURE
INTENSITY
MORPHOLOGY
SEQUENTIAL COMBINATION
AUTOMATION LEVEL
MANUAL
SEMI-AUTOMATED
FULLY-AUTOMATED
EVALUATION OF REGISTRATION RESULTS
ERROR METRICS
GROUND TRUTH METHODS
REGISTRATION RESULTS
REGISTRATION DECISIONS
UNCERTAINTY:
-GLOBAL VS LOCAL FIT
-MODEL OVERFITTING
-INVARIANCE DEGREE
-LIMITED PARAMETER SEARCH SPACE
-ROBUSTNESS TO MODEL DEVIATIONS
-GOODNESS OF EVALUATION CRITERIA
COMPLEXITY OF REGISTRATION ALIGNMENT
IMAGE SUBAREA RIGID MODEL VISUAL INVARIANCE MANUAL METHODS
ENTIRE IMAGE AFFINE, ELASTIC MODEL COMBINED INVARIANCE SEMI- & FULLY-AUTOMATED METHODS

Figure 5.
Registration decisions and their impact on registration results.

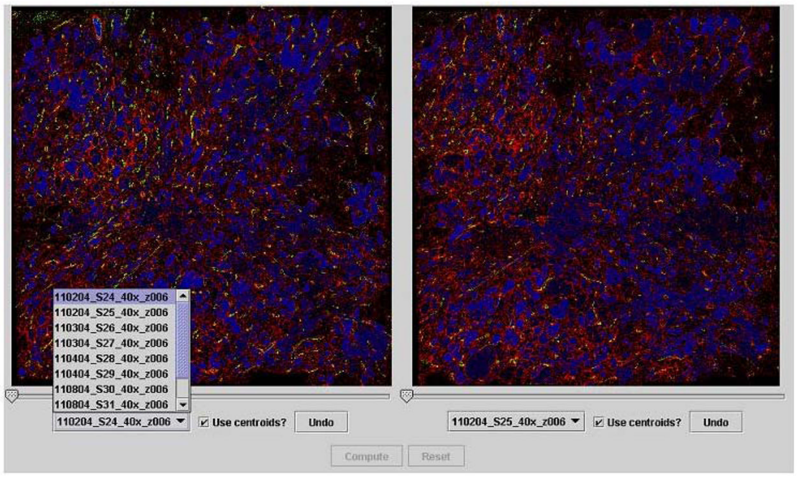


Figure 6.
Image selection menu.

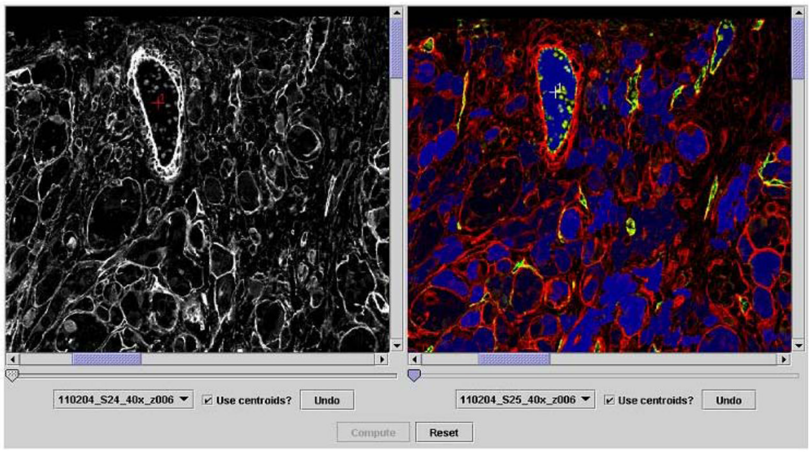
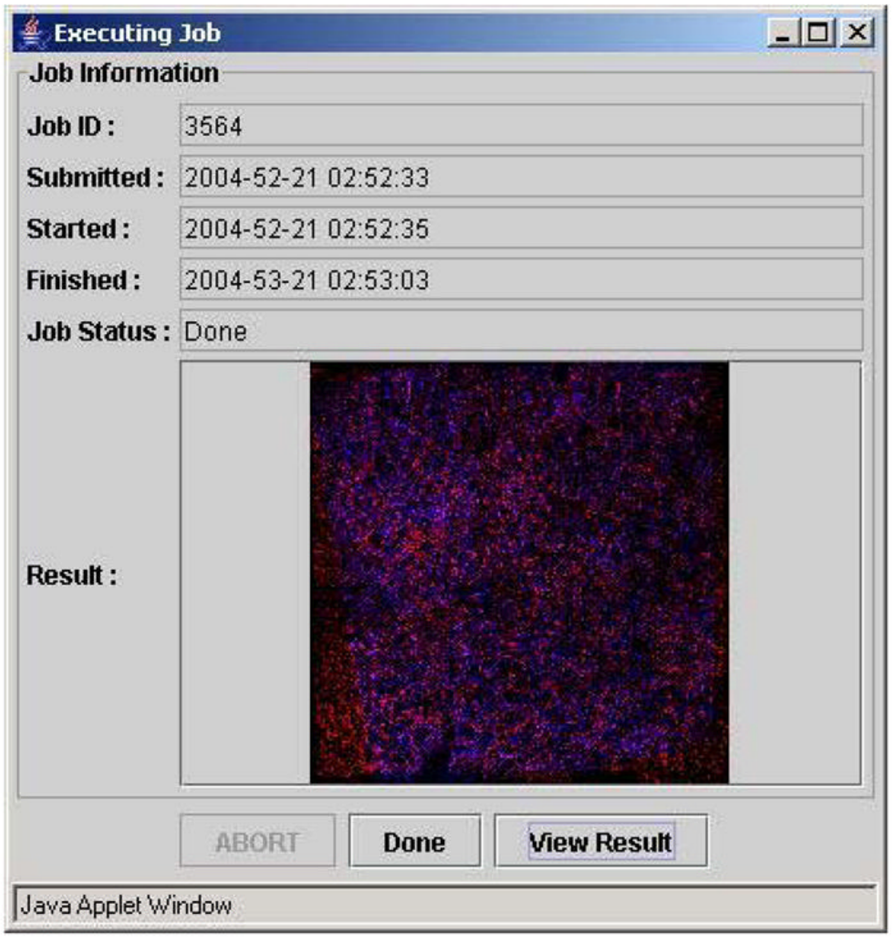


Figure 7.
Selection of registration points using the centroid feature approach. Note that sub-areas the large images in Error! Reference source not found. are shown. ("+" marks represents the region centroids)



Job ID : 3564
Submitted : 2004-52-21 02:52:33
Started : 2004-52-21 02:52:35
Finished : 2004-53-21 02:53:03
Job Status : Done
Result :

Figure 8.
Job execution window after pressing the button “View Result”.

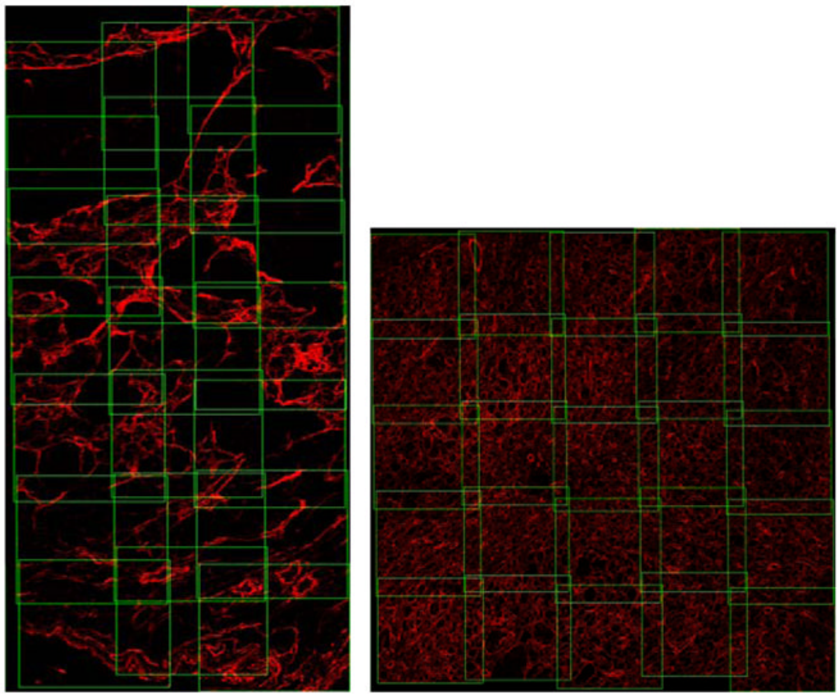


Figure 9.
The images after mosaicking. The edges of each tile are visualized using green lines. On the left is a frame from dataset 1 and on the right is a frame from dataset 2.

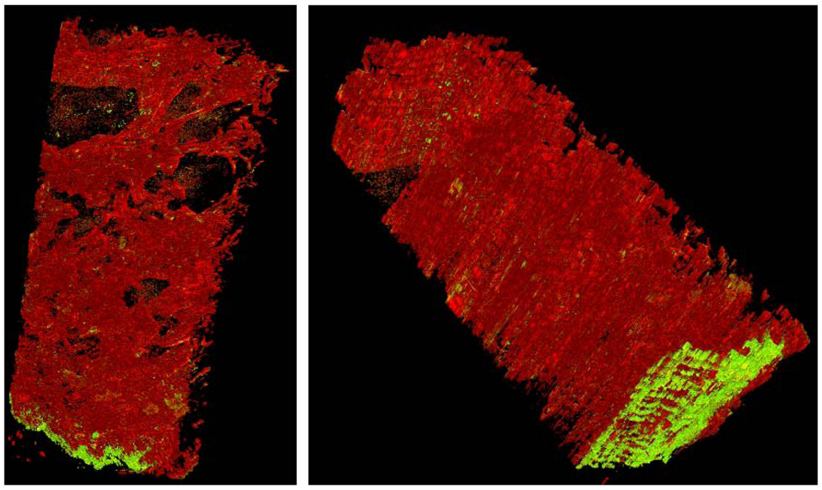


Figure 10.
3D view of the reconstructed 3D volume.

Table 1

Timing of sending a packet from an application to the server and receiving an answer back in ms using UDP, TCP and D2K Web Services

	UDP[ms]	TCP[ms]	Web Services[ms]
Application local	0.35	1.11	9659.89
Application remote	9.44	19.41	9598.13

Table 2

Tradeoffs considered during a prototype system design using web services for 3D volume reconstruction. C refers to a constant, + indicates increase and − is decrease. Multiple + or − symbols indicate the magnitude of the value specified in each column.

	Client CPU	**Client RAM**	**Bandwidth**	**# of Queries**
Image pyramid	C	−−	++	+
Client does segmentation	+++	−	−	C
Compression	+	+	−−−	C
Transformation on client side	+++	+	−	−

Table 3

Datasets

	Dataset 1	**Dataset 2**
Tile Size (pixels)	512 × 512	512 × 512
Tiles per frame (row × column)	7 × 3	5 × 5
Frame size after mosaicking (pixels)	2746 × 1170	2300 × 2300
Frames in a subvolume	13	13
Subvolumes	16	10
Channels	2	2
Memory Size	1336.5 Mbytes	1375.4 Mbytes