

Viewing the Visible Human using Java and the Web

Yuh-Jye Chang, School of Computer and Information Science, Syracuse University, Syracuse NY 13244, U.S.A.

Paul Coddington, Northeast Parallel Architectures Center, Syracuse University, Syracuse NY 13244, U.S.A. and

Department of Computer Science, University of Adelaide, Adelaide SA 5005, Australia (current address), paulc@cs.adelaide.edu.au.

Karlie Hutchens, Department of Computer Science, University of Adelaide, Adelaide SA 5005, Australia.

6 July 1998

Abstract

The Visible Human is a large data set of digitized images of a male and female human body. We discuss the issues involved in providing users with remote access to this data using the World Wide Web. In particular, we describe the design and implementation of a Web-based viewer for the Visible Human images, using a Java applet interface to enable the user to select the type of data and the desired sectional slice. We also investigate additional capabilities that can be provided by utilizing server-side programs and databases. Many of the issues discussed are relevant to general Web-based digital multimedia repositories.

Keywords: Java, Visible Human, digital libraries

1 Introduction

The Visible Human Project [1] of the U.S. National Library of Medicine has created digital image data sets of a human male and female for use in education and research. These data sets provide an interesting example of a large (tens of Gigabytes) digital image library. The Visible Man, a set of digital images of two-dimensional slices of a male cadaver, became available in 1994. The Visible Woman followed in 1996.

Many educational products have been produced using the Visible Human data set, particularly in CD-ROM form. Our approach has been to make the images available online to anyone with an Internet connection and a Web browser, which has become a standard interface for accessing digital information. One of our main goals was to make the images available for educational use [2], so the system had to have a fast, responsive, intuitive, and user-friendly interface to the images. In this paper, we discuss the design and implementation of an online repository of Visible Human image data, and a Java applet

interface [3] for remotely accessing the images using a Web browser. Many of the issues discussed here are of more general relevance to accessing digital multimedia libraries using the Web.

A major consideration in designing a Web-based digital image library is that users may be accessing the images using a wide range of computers and network bandwidths, from low-end PCs with modem connections to high-performance workstations with a high-speed network connection. The memory and processing speed can vary by a factor of 10 or more, while the bandwidth can vary by a factor of 1000. For many clients, the screen size may only be 640x480 pixels, which can be a substantial limitation in designing a graphical user interface. Our goal in the design and implementation of the viewer was that it should be usable over the widest possible range of computer and network capabilities using a standard Web browser.

At the time we started the project, the only way to provide a Web interface to allow the user to select an image for viewing was either through a Common Gateway Interface (CGI) program accessed via an HTML form that allowed the user to type in a slice number, or through a clickable image map that allowed the user to click on the desired area of an image map of the body. Creating something more interactive required building a client/server program. Such programs have been developed for viewing the Visible Human images [4,5], but they are generally limited to a particular operating system. However in order for the program to be used in schools, the client program would have to run on both Macintosh and PC.

The development of the Java [6] programming language enabled a much better way of implementing the image viewer, by using an applet [7] which can be run within the Web browser, and can readily access multimedia information from a Web server. Java is particularly suitable for interactive, multimedia educational applications, since it eliminates software portability, installation and maintenance problems, which is particularly important for schools, which have limited resources and staffing.

2 The Java Visible Human Viewer

Our Visible Human Viewer [2] is a Java applet which is accessed from a Web page that provides information on how to use the applet, and links to additional information. Version 1.0 of the applet provided access only to the Visible Man, since the female images were not yet available. The applet was recently upgraded to version 2.0, which has several additional features (see section 3) and an interface to the Visible Woman data.

2.1 User Interface

The main task in creating the image viewer was designing and implementing an intuitive and easy-to-use graphical user interface for accessing a large set of image data from a Web server. One of the major advantages of using a Java applet for the interface is that it allows great flexibility in the design, as well as improving performance by allowing many operations to be done on the client-side, thus minimizing latency overheads by eliminating unnecessary calls to the server. The interface is quite similar to those used in some CD-ROMs of Visible Human images, although it has some special features due to the use of the Web. The Web provides some advantages over CD-ROMs, such as a standardized client/server interface and the ability to access larger, and more up-to-date, data sets. It also has some disadvantages, such as being restricted to the existing Web standards (such as browsers, languages and

APIs) and the relatively large overhead in having to access this data over a wide-area network. For large image libraries such as the Visible Human, a Web interface allows the user to access more data than could fit on a CD-ROM, but the data may have to be downloaded very slowly or at lower resolution if the user has a slow network connection.

The viewer interface is shown in figure 1. The applet begins by showing small thumbnail (or preview) images of slices through the human body in three different cutting planes - axial, sagittal and coronal. Each image has a *cutting line* running across it, which the user can move (by clicking and dragging the mouse) to the position where they wish to view a slice of the body. After the cutting line has been moved, a thumbnail image of the requested slice is downloaded from the server and displayed.

The user can choose between the male and female data set, and the interface also supports the choice of different types of images (such as MRI), although currently only photographic images have been processed and made available. When the user finds a thumbnail image they want to study in more detail, they can select the desired resolution, and then download the image by clicking the `Load Image` button. The applet then creates a pop-up window for viewing the image, as shown in figure 2. At the higher resolutions, the images are larger than the size of the computer screen, so the pop-up window displaying the image allows the user to pan, scroll and resize the image.

2.2 Processing the Image Data

The time taken to download images can vary enormously depending on the speed of the network used to access the images. We therefore provide a choice of 3 different image resolutions (in addition to the thumbnails): the highest is the full resolution of the Visible Human dataset, the lowest is suitable for downloading over a fast modem, and the medium resolution lies somewhere in between. Providing multiple resolution images is crucial for Web-based access to a digital image library, not just because of the widely varying network bandwidths for different clients, but also because low-end clients may not be able to handle the memory requirements of very large images.

In order to provide acceptable download times for the images, JPEG compression was used, which reduced the file size by a factor of about 20, while still retaining reasonable image quality. Pre-processing of the image data, including cropping, image processing, compression, and creating multiple resolution data sets, took many weeks of effort and hundreds of hours of computer processing time, however it allowed us to reduce the size of the uncompressed raw data set by a factor of almost 100. This was crucial to providing a Visible Human viewer that was usable with the bandwidths currently available for Web browsing.

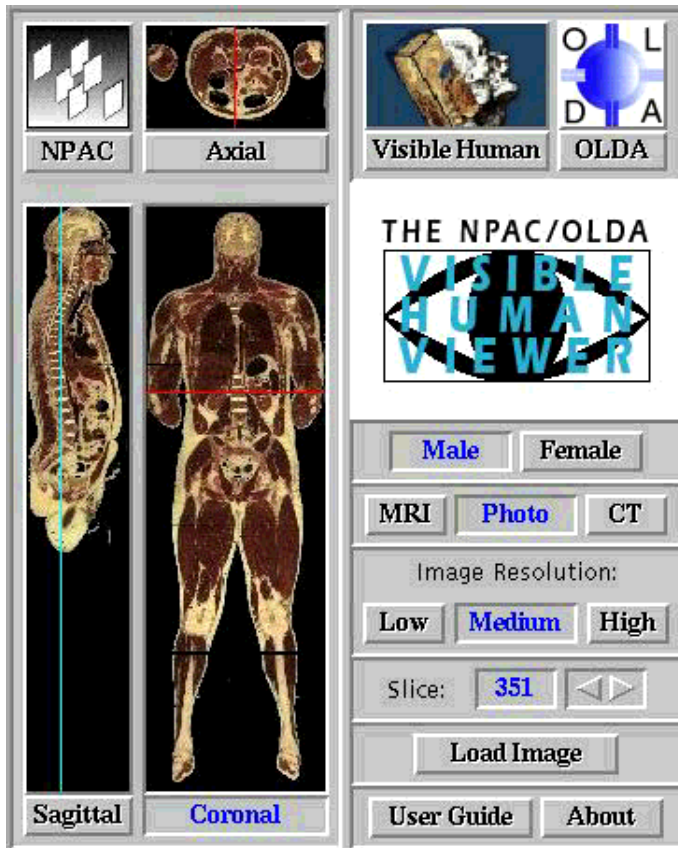


Figure 1: The main panel of the Visible Human Viewer applet. The active cutting line is visible in the sagittal image, producing the coronal slice displayed in the neighboring window.

2.3 Problems in using Java

Several problems were encountered in the use of a Java applet for the viewer, however we were able to find workarounds for most of them.

In principle, Java offers the huge advantage of portability across all platforms, or "write once, run anywhere". However in practice we encountered many portability problems, with both the Java Development Kit (JDK) used to create the program, and the Java Virtual Machine (JVM) used to run it. A lot of effort was expended in creating workarounds for problems that occurred in different JVMs, particularly in the early implementations within Web browsers. It was a common occurrence that the Java applet would work fine in the JDK appletviewer, and in a particular version of the Web browser for a particular operating system, but part or all of the program would not work for a different version of the browser on a different operating system. Even with the most recent versions 4.0.x of Netscape Communicator, the applet works for some values of x and not others. However, this is mostly a maturity issue - Java is a new and evolving language, so its early implementations can be expected to be buggy.

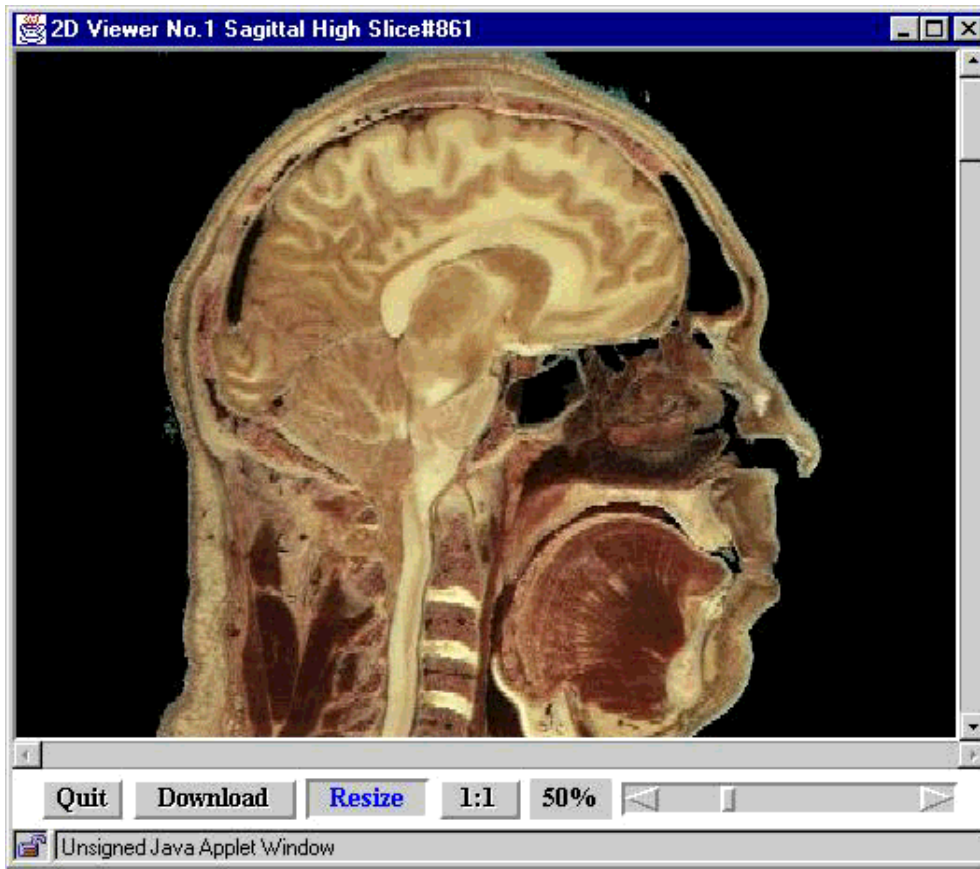


Figure 2: The pop-up image window for displaying a downloaded image.

A related portability problem is that user interfaces built using the Java Abstract Windowing Toolkit (AWT) can have a different look and feel on different operating systems or JVMs, and AWT components (buttons, scrollbars, text, etc) can have different relative sizes. We wanted to keep a consistent-looking interface, and spent considerable effort developing our own interface components so they would look the same on all clients. This problem has since been addressed by the Swing GUI toolkit, which is part of the Java Foundation Classes available with JDK version 1.1.

A possible disadvantage of using a Java applet, rather than an application, is that the program has no access to the local file system. This means that the user cannot directly save an image file of interest using the applet program. However, since the data files are accessed by the Java applet as URLs, the applet can give the URL to the browser or an image viewer helper application for display. In the applet, this is done using the `Download` button in the pop-up window displaying the image. The user is then able to save the image file from the browser or image viewer application. This is an advantage of accessing the data directly as a URL, rather than through a socket connection, for example. Also, since the image was accessed as a URL, the browser knows that the image file is available in the cache, so it does not need to download it again.

A disadvantage of using Java applets to view images is that Java (and Web browsers) tend to use rather poor dithering algorithms to convert the 16 million (24-bit) color palette of the images to the 256 (8-bit) colors available on most computer screens. The images generally look much better when displayed using a specialized image viewing program.

Java stores images in an inefficient way, using 32-bit integers rather than 1 byte for grayscale images or 3 bytes for color images. This means that large images can have problems with memory limitations. Many JVM implementations have problems with using virtual memory, so accessing large images can cause the JVM (and the browser) to run extremely slowly or to crash.

3 Adding Functionality using the Server

There is plenty of scope for enhancing the Visible Human Viewer to make it more effective as a learning tool. Here we discuss some enhancements that add useful functionality to the applet, but require the invocation of programs running on the server. This can be done using either CGI or the more recent Java Remote Method Invocation (RMI) [8].

3.1 Selecting a Region

The high-resolution sagittal and coronal images are very large and can take a long time to download. This is very wasteful if the user is only interested in looking at a small section of the image. An obvious solution to this problem is to allow the user to select a particular region of the image. This can be done by drawing a rectangular box on the desired preview image using the standard "rubber-banding" method.

Implementing this capability requires a server-side program that will uncompress the image, crop out the desired region, JPEG compress the resulting cropped image and return it to the client applet for display. We have implemented two versions of this, one using a perl CGI program and one using Java RMI. Since RMI can only be used by more recent browsers that support JDK 1.1, and is much slower than using CGI, the current version of the viewer uses CGI to do the cropping. The CGI program is quite fast, taking less than a second even for the highest resolution images.

3.2 Animations

A number of groups have created animations of Visible Human images (see the Visible Human Project [1] for a list). These can be manipulations (e.g. rotations or fly-throughs) of 3D reconstructions, or just moving through a section of the body by animating a collection of 2D image slices. It is straightforward to create animations from successive 2D images, and we have created a number of these animations in MPEG format and made them available on the Web [9]. As with the image slices, the animations have been created at different resolutions, so they can be downloaded and played on a range of clients.

It would be possible to allow the user to create their own animations of areas of interest. The user could select a rectangular region of interest and a group of slices to be animated, and a program on the server could use this information to generate the input file for the MPEG encoder, run the encoder, and return the video stream. Creation of MPEGs is extremely CPU-intensive, and currently this would produce an undesirably large computational burden for the Web server and waiting time for the user. However in the near future, faster computers, faster networks, special-purpose chips for MPEG encoding, and standards for streaming video, will enable real-time interactive creation and viewing of animations of this kind over the Web.

3.3 Anatomical Database

A useful additional capability would be to link the Visible Human images to information about the human body. For example, a database of names of anatomical features (bones, muscles, organs, etc) could be accessed via the applet, enabling the user to click on an area of an image and find out what they are looking at, or to search for a particular feature and have the program return a set of images highlighting that feature.

A database of this kind, which maps pixels of the images to names of anatomical features (and vice versa), is currently being developed by the Visible Human project. When it becomes available, the database could be made accessible on the Web server, and the applet could be augmented to query the database and return appropriate information. An advantage of a Web-based system is that anatomical information of this kind could be augmented with hyperlinks to more detailed information available on the Web.

3.4 Teaching and Collaboration Tools

Interfacing the applet with a server program allows the sharing of state information between multiple instances of the applet on different clients. For example, this would allow a teacher to run the viewer applet to illustrate certain anatomical features, and have the same interface and images be displayed on the computers of all the students in the class, if they were running the same applet program connected to the server. This approach may be useful for distance learning and remote collaborative work. The Visible Human Viewer has been ported to two systems of this kind, Habanero [10] and TANGO [11]. Both of these systems require a server program running in conjunction with the Web server, and minor modifications to the applet program to allow it to exchange state information with the server. TANGO also provides shared whiteboard and text-based chat facilities, to allow further interaction.

4 Conclusions

Our Visible Human Viewer first came online in December 1995, soon after the first Java-enabled Web browsers became available. It was one of the first Java applets to provide a fairly sophisticated, interactive graphical user interface that enabled the user to access multimedia information from a large digital library on a remote server. Subsequent improvements provided additional functionality such as cropping of images and collaborative viewing, using server-side processing. The viewer illustrates the power of Java and the Web for use in interactive access to online digital multimedia libraries, particularly for educational applications.

Acknowledgements

We would like to thank Geoffrey Fox for suggesting and supporting the project, and Rick Rodgers of the National Library of Medicine for helpful suggestions to improve the user interface for the program. Thanks also to Alvin Leung and Kelly Pellini for helping with some of the data processing.

Initial work on this project was supported by the Northeast Parallel Architectures Center (NPAC) at Syracuse University. Further work was supported by the On-Line Data Archives (OLDA) project of the

Research Data Networks (RDN) and Advanced Computational Systems (ACSys) Cooperative Research Centers. RDN and ACSys are established under the Australian Government's CRC Program.

References

[1] The U.S. National Library of Medicine. The Visible Human Project.
http://www.nlm.nih.gov/research/visible/visible_human.html.

[2] Kim Mills *et al.*. The Living Textbook and the K-12 Classroom of the Future. *Proceedings of SuperComputing '95*, San Diego, CA, November, 1995.
http://www.supercomp.org/sc95/proceedings/620_KMIL/SC95.HTM.

[3] Paul Coddington, Yuh-Jye Chang and Karlie Hutchens. The NPAC/OLDA Visible Human Viewer.
<http://www.dhpc.adelaide.edu.au/projects/vishuman/>.

[4] C. North, B. Shneiderman, and C. Plaisant. The HCIL Visible Human Explorer.
<http://www.cs.umd.edu/projects/hcil/Research/1995/vhe.html>.

[5] C. North, B. Shneiderman, and C. Plaisant. User Controlled Overviews of an Image Library: A Case Study of the Visible Human. Proc. of ACM Digital Libraries '96, ACM Press, 1996.
http://www.cs.umd.edu/projects/hcil/Research/1995/vhp/dl96/vhe_dl96.html.

[6] Sun Microsystems Inc. The Java Platform. <http://java.sun.com/aboutJava/index.html>.

[7] Sun Microsystems Inc. Applets. <http://java.sun.com/applets/index.html>.

[8] Sun Microsystems Inc. Java Remote Method Invocation (RMI).
<http://java.sun.com/products/jdk/rmi/index.html>.

[9] Paul Coddington. Visible Human Animations.
<http://www.dhpc.adelaide.edu.au/projects/vishuman/mpegs.html>.

[10] National Center for Supercomputing Applications. The Habanero Project.
<http://www.ncsa.uiuc.edu/SDG/Software/Habanero/>.

[11] Northeast Parallel Architectures Center at Syracuse University. The Tango Project.
<http://www.npac.syr.edu/projects/tango/>.