

A Web-Based Solution for Viewing Large-Sized Microscopic Images

Chung-Yueh Lien,¹ Hsu-Chih Teng,² Deng-Ji Chen,² Woei-Chyn Chu,¹ and Chia-Hung Hsiao³

This paper demonstrates a pure web-based solution enabling the presentation of scanned pathologic microscopic images on the web. For each slide, an entire specimen is scanned, and a high-resolution digital image (in the order of giga-pixels) is reconstructed. These huge images are then tiled into many 256×256-pixel blocks with different resolutions, and information about the blocks of each scanned slide is included in an extensible markup language metafile. Based on the data, a virtual microscopy system is created for viewing the scanned pathologic slides on web. The functionalities (changing viewing resolution, location adjustment, and multimedia annotation presentation) of our virtual slide viewing system are accomplished using pure hypertext markup language (HTML) and JavaScript. We show that there is no need to add plug-in components to browsers in order to handle virtual slides on the web. In a heterogeneous healthcare environment, methods using pure HTML and JavaScript to deal with pathologic content are more appropriate than using proprietary technologies supported only by specific browsers.

KEY WORDS: Virtual microscopy, virtual slide, JavaScript, XML, whole slide imaging

Abbreviations:

DICOM digital imaging and communication in medicine
IHE integrating healthcare enterprise
VS virtual slides
VSAS virtual slide acquisition systems

INTRODUCTION

Digital imaging and communication in medicine (DICOM) is a successful protocol for accessing and viewing digital radiologic images. Although DICOM defines the object of microscopic images, converting pathologic images into DICOM objects and viewing the images in a conventional DICOM viewer are not suitable for

pathologic diagnosis. In pathology, a whole view of specimens is required for diagnosis. Pathologists need to be able to navigate whole specimens on a computer system as virtual slide (VS) that a slide can be manipulated under microscope and enlarge portions of the sample that they feel might be interesting. Since 1998, several VS systems have been developed by different commercial and academic bodies.¹ These systems are currently being used not only to improve the teaching and learning efficacies of histology and pathology in medical school education²⁻⁷ but also for telemedicine, academic conferences, and workshops,^{8,9} they could also be used in clinical diagnosis if quality assurance, image quality, and hardware/software were improved.^{1,9-12} To achieve this aim, complete slides of specimens must be scanned and digitized using a high-resolution microscope, the results of which are very large digital images (in the order of gigabytes for each slide). Images of this size are too large to be handled by conventional DICOM

¹From the Institute of Biomedical Engineering, National Yang Ming University, Taipei, Taiwan.

²From the Institute of Computer Science and Engineering, College of Computer Science, National Chiao Tung University, Hsinchu Taiwan.

³From the Department of Medical Informatics, Tzu Chi University, No.701, Sec. 3, Jhongyang Rd., Hualien City, Hualien 970, Taiwan.

Correspondence to: Chia-Hung Hsiao, Department of Medical Informatics, Tzu Chi University, No.701, Sec. 3, Jhongyang Rd., Hualien City, Hualien 970, Taiwan; tel: +886-3-8565301; fax: +886-3-857-9409; e-mail: chhsiao@mail.tcu.edu.tw

Copyright © 2008 by Society for Imaging Informatics in Medicine

Online publication 27 June 2008
doi: 10.1007/s10278-008-9136-x

viewers, and the transfer of such large images through DICOM network protocols involves large overheads. The large data size means that following current DICOM specifications to package the whole scanned slide data as a DICOM object is not appropriate.

Although the size of the reconstructed pathologic images is very large, only a small portion of the images can be displayed on a computer screen at one time (limited by the screen size and resolution of the computer system), and it is not necessary to transfer the whole image to a client through the network when we wish to examine it. Based on this concept, many virtual slide image systems have been constructed and demonstrated, all of which use web-based solutions to handle the interaction between server systems and remote clients. A browser is used to navigate the pathologic images in these systems. A web-based virtual slide solution is ideal for enabling viewing of pathologic images on a network, and an hypertext transfer protocol (HTTP) link enables easy system integration. However, most virtual slide systems use proprietary protocols and plug-in components (such as Java Applets or Microsoft ActiveX controls) to transfer and display pathologic images; these plug-in components cannot be handled by other systems, and it is difficult for us to use proprietary protocols for accessing pathologic images that are stored on the server. This constriction has

limited the expansion of the virtual pathologic slide systems for other uses such as integration with reporting systems for clinical diagnosis or using virtual slide images as content in standard learning systems for medical education.

In this paper, we use pure hypertext markup language (HTML) and JavaScript to construct a virtual slide image viewing system and use HTTP protocol to transmit digital microscopic images. We show that the fundamental requirements of a virtual slide image viewing system can be accomplished using a pure web-based solution; we also demonstrate that the pure HTML and JavaScript solution can easily be expanded for use with other web systems, such as a web-based pathology report system, and can be utilized by various browsers and web servers, which is very important in a heterogeneous healthcare environment.

METHOD

Data Structure of Digitized Microscopic Slides

Figure 1 illustrates the architecture of the virtual slide system. Virtual slide images are acquired using the slide digitizer, and doctors can use authoring tools to generate annotations on the scanned slide images. Each whole slide image is

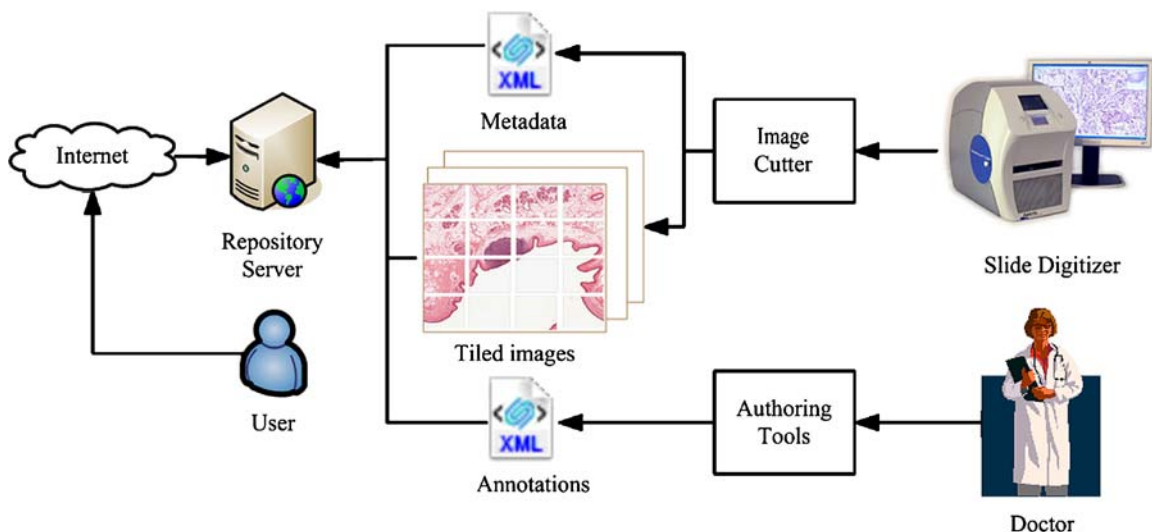


Fig 1. Architecture of the virtual slide system.

cut into lots of small tiled images, for which extensible markup language (XML) annotations are stored in a VS repository server, a web server that stores VS data and the corresponding HTML and JavaScript templates for presenting the VS data. Consequently, a browser can be used to connect to the repository server in order to view entire VS images.

To construct a virtual slide of a pathologic specimen, the entire slide must be scanned using a high-resolution microscope. In general, commercial virtual slide acquisition systems can be divided into two main categories: video camera and still camera. The video camera method uses linear-array detectors to create seamless complete slide images, and the still camera method uses a fixed-area camera to capture thousands of individual image tiles and joins them together to create a large mosaic of a single slide.^{1,13} Dedicated methods such as neural networks or global geometric and radiometric transformation have been developed in order to correct overlapping areas and/or other image artifacts.¹⁴ In this research, glass slides are digitized using an Aperio ScanScope (Aperio Technologies, Inc., CA, USA) slide digitizer, which combines linear-array detectors with an objective lens for scanning slide samples. The objective lens resolution can be changed in the same way as a conventional microscope. This study used a $\times 20$ objective lens to acquire a high-resolution (giga-pixel) digital image of each slide.

Each slide scan can be saved as a single file. However, the image is too large to process directly; it would take a long time to transmit the huge amounts of data through the network, and the original image is too large to be handled by browsers. In order to distribute the VS data on the internet, we divide the scanned image into many small image blocks (256×256 pixels each), which are small enough to be handled by computers and network systems. To further reduce the response time, lower-resolution image blocks are also generated. The blocks of a scanned image are prepared and stored at different resolutions, for example, if the slide is scanned using a $\times 20$ objective lens in the microscope scanning system, not only are image blocks of the original resolution prepared but lower-resolution blocks ($\times 10$, $\times 5$, and $\times 2.5$) are also generated from the scanned result (Fig. 2). Each block is saved as a single file, and

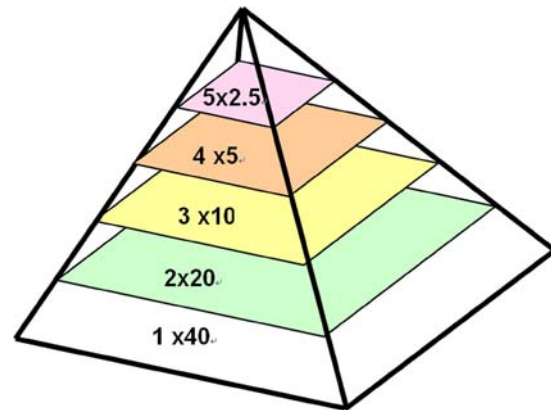


Fig 2. Pyramidal resolutions of tiled blocks.

block information such as file name, resolution, and orientation is stored in an XML metafile. When we want to view a specific region of a virtual slide at a certain resolution, the web server loads the particular blocks within the region and reconstructs the image corresponding to the region. Because the size of an image that can be displayed on a browser is limited by the screen size, each time a browser requests a specific region of the scanned image, we only need to load a small number of blocks to display on the browser. This greatly reduces the response time of our web server. The architecture of the virtual slide data is defined as in the following virtual slide XML metafile (Fig. 3).

Web Display of Virtual Slides

Based on the tiled blocks and the virtual slide XML metafile, it is convenient to use pure HTTP protocols and JavaScript to navigate huge images on the web. We define a canvas in our virtual slide web page: The web page supports a JavaScript functionality, which allows the mouse to be used to drag and drop the virtual slide image displayed on the canvas, moving the display canvas relatively over the huge virtual slide image (Fig. 4).

In Figure 4, the background is a huge microscopic image that was tiled as fixed-size blocks (Bi0j0, Bi0j1..., Bimjn). In a virtual slide system, the display canvas may navigate freely over the huge image; in a web system, we can use JavaScript running on a browser for displaying the tiled blocks to achieve the moving effect. As demonstrated in Figure 4, P0, P1, and P2 are the

```

<VirtualSlideImageObject>
<GeneralDescription> <!-- General description of the virtual slide object -->
    <SeriesUID>12.34.55.22.1</SeriesUID> <!--series instance UID of the slide -->
    <ObjectUID>12.34.55.1203</ObjectUID> <!--Service Object Pair instance UID of the slid , the whole slide has one
    ObjectUID -->
    <ObjectiveLensResolution>20 </ObjectiveLensResolution> <!--Objective lens resolution for sampling the slide -->
<TotalRows>40448</TotalRows> <!--Total rows of the whole slide -->
    <TotalColumns>52480</TotalColumns> <!--Total columns of the whole slide -->
    <SaveImageType>jpeg</SaveImageType> <!--Save image type of each block of the slide -->

</GeneralDescription>
<ResolutionLevels> <!--The descriptions of different Resolution Level -->
    <ResolutionLevel><!--The descriptions of current level -->
        <RescaledResolution>20</RescaledResolution><!--Rescaled Resolution of current level -->
        <ImageBlockRows>256</ImageBlockRows><!--Rows of each image block-->
        <ImageBlockColumns>256</ImageBlockColumns><!--Columns of each image block-->
        <RowCount>158</RowCount><!--Number of blocks in raw of current level-->
        <ColumnCount>205</ColumnCount><!--Number of blocks in column of current level-->
        <Path>\20\</Path><!--Path for storing each block of current level-->
        <BasicFileName>Block20</BasicFileName><!--Basic file name of each block. Block file name would be
        generated according the basic file name and the number of row and column of the block. For example, the file of
        block(Row:3 ,column:0) of current level would be Block20_3_0.jpg. -->
    </ResolutionLevel>
    <ResolutionLevel> *** </ResolutionLevel>
    <ResolutionLevel> *** </ResolutionLevel>
    <ResolutionLevel> *** </ResolutionLevel>
</ResolutionLevels>
</VirtualSlideImageObject>

```

Fig 3. XML metafile describing a virtual slide image object.

upper left points of the canvas when we move the viewing position. When a mouse move event with the movement (dX0,dY0) is initiated, the canvas moves from P0 to P1. In this case, the blocks

(Bi0j0, Bi0j1 Bi0j2, Bi0j3, Bi1j0, Bi1j1, Bi1j2, Bi1j3, Bi2j0, Bi2j1, Bi2j2, Bi2j3) that should be displayed on the canvas are the same; only the position of each block will move (-dX0,-dY0).

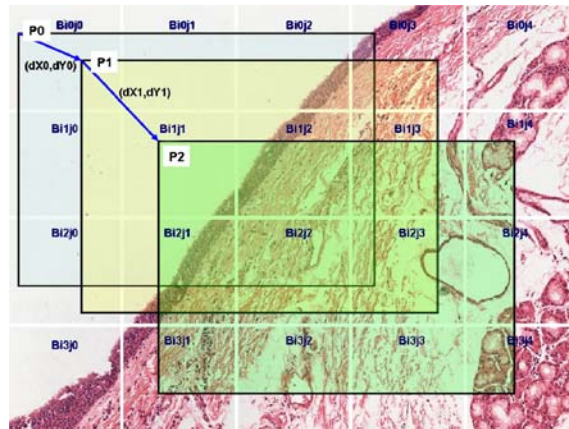


Fig 4. Graphic demonstrating a display canvas moving on a huge background image.

When the canvas moves from P1 to P2, the blocks (Bi1j1, Bi1j2, Bi1j3, Bi2j1, Bi2j2, Bi2j3) are still on the canvas with the position adjustment ($-dX1$, $-dY1$), and new blocks (Bi1j4, Bi2j4, Bi3j1, Bi3j2, Bi3j3, Bi3j4) are also shown on the canvas. The position adjustment of blocks and addition of new blocks to the canvas can be accomplished using a JavaScript function that corresponds to the movement event for viewing the VS images. When the canvas moves from P0 to P1, we only have to adjust the position of blocks that have already been loaded to the browser: The JavaScript event function runs very quickly, and the VS image shown in the canvas moves smoothly. When the canvas moves from P1 to P2, the JavaScript function should initiate HTTP requests and acquire new blocks from the web server. If the blocks are small enough in size (e.g., an average of 40 kb per block, as in our system) and the network bandwidth is acceptable, the process of adding and displaying new blocks on the canvas can also be achieved within a fraction of a second. As described above, we can construct a web-based virtual slide viewing system without limiting efficiency by using only simple HTML and JavaScript.

The Presentation of Annotations on VS Images

Annotations marked on the virtual slide images can also be presented by pure HTML and JavaScript only. The following XML data demonstrate an annotation that was generated by a virtual slide editor. Based on the XML data, we can also

display the annotation on the virtual slide web page.

As in Figure 5, the shape, color, and stroke of each annotation are defined in the XML tags, and we use `<Points>` to define the position, orientation, and size of an annotation on the virtual slide image plane. When viewing virtual slides on the web, the XML data (virtual slide XML metafile and corresponding annotation XML file) are loaded to the browser. When we move the viewing canvas on the virtual slide image plane, we also adjust the locations of the annotation that should be displayed on the canvas. In the JavaScript event corresponding to canvas movement, we estimate the location of each annotation relative to the position of the current canvas and redraw the annotations that should be displayed on the canvas in the JavaScript event. The graphic drawing functionalities are based on a pure JavaScript Vector Graphics Library.¹⁵ Using the functions provided by the library, graphics (line, triangle, ellipse, polygon...) can easily be shown and redrawn on a web page. As we use pure JavaScript to handle the presentation of virtual slide images and annotations on the browser, the virtual slide viewing system functions properly in most browsers (IE, FireFox, Opera, etc.).

Selecting a Region of the Virtual Slide Image

We usually want to select regions of virtual slide images and display the evidence regions on pathology reports. This requirement can easily be accomplished in our virtual slide viewing system.


```

<Annotations>
<ReferencedObjectUID>12.34.55.1203</ReferencedObjectUID> <!--Referenced virtual slide image UID -->
<AnnotationsUID>12.34.55.1203.0.1</AnnotationsObjectUID><!-- The annotations' UID -->
<Annotation>
  <AnnotationID>1</AnnotationID><!--Current annotation's ID -->
  <Attribute>Matured adipocytes</Attribute><!--Attribute for describing the annotation -->
  <Shape>Arrow</Shape><!--define shape of the annotation, exp. line, triangle, ellipse, polygon...-->
  <Color>888800</Color><!--rgb value, exp. r=88 g=88 b=00 yellow -->
  <Stroke>4</Stroke><!--Stroke of the shape -->
  <Points><!--Points for defining the shape, number of point depends on the shape. For example, line or arrow 2
    points, triangle and ellipse 3points, polygon n points...-->
    <Point X="27557" Y="27222"/><!-- X Y values of the point on virtual slide image plane-->
    <Point X="27782" Y="26962"/>
  <Points>
</Annotation>
<Annotation> ***</Annotation>
<Annotation> *** </Annotation>
</Annotations>

```

Fig 5. XML annotation file containing annotations of a virtual slide image.

A rectangular region in the image plane of a virtual slide can be defined by two vertex points of the rectangle. For example, we set the two vertex points of a rectangular region as (X1, Y1) and (X2, Y2). These points are assigned by our user interface (Fig. 6). Usually, only a limited number of blocks would be required to reconstruct the region (selecting low resolution with large area or high resolution in small area of whole slide). There are two ways to display the region in a pathology report web page. The first is to recombine the blocks into a jpeg image and cut the edge of the image out of the region. The reconstruction process is performed in our web server, and the cut jpeg image is posted to the pathology report web page. The second method is setting a canvas in the pathology report web page. The size of the canvas is assigned as the same as the region that we want to display, and the metadata that describe the location of each block that should be shown on

the canvas are also included in the web page. When we open the web page, the blocks within the region will be loaded to the browser. With the metadata and block images, we can redraw the virtual slide region using a method similar to the presentation of virtual slides on the web (as in Fig. 4).

RESULTS

Virtual Slide Viewing System for Medical Education

As described above, we have constructed a virtual slide microscopic image viewing system, which is used to assist in the histology and pathology education of medical students in our university. Students can use the system, instead of a conventional microscope, to examine the normal and abnormal cell structures of slide samples.



Fig 6. A region (X:256, Y:140, X:673, Y:468) is defined by mouse move events provided in our virtual slide system.

Unlike with a conventional microscope, it is easy to identify the location of a specific structure on digital virtual slides. We have successfully used the virtual slide-assisted teaching system at the National Yang Ming University (NYMU) from 2006 in year 3 pathology and laboratories courses for 179 medical and dentistry students, and in 2007 fall semester, the 3-year histology course was added. There are up to 200 slides of samples of various organs used in the teaching. Students on courses and in the teaching laboratory can carry their notebook and use the campus wireless LAN to link to the virtual slide website (<http://140.129.165.162/moodle/index.php> in Traditional Chinese) using moodle (<http://moodle.com/>) to manage the online courses. Students can use the system, instead of a conventional microscope, to examine the normal and abnormal cell structures of slide samples. Unlike a conventional microscope, it is convenient to identify the location of a specific structure on the digital VS, and we have added annotations of histology and pathology descriptions to each slide (Fig. 7).

Linking VS Images with Pathology Reports

Our virtual slide system can easily integrate with HTML-formatted pathology reports. As described in the “Method” section, we can display the evidence regions of virtual slide images on a pathology report. The region can be selected in our virtual slide viewing user interface (Fig. 6); then, after the region has been defined, our system will automatically generate an XML file (XY offset values, resolution and blocks in the region) to define the region. Accomplished using the XML file, the server has a JavaScript function with an HTML template for displaying the region and adding the evidence report (Fig. 8). The report web page presents patient demographics, illness history, clinical diagnosis, and images of the entire tissue specimen and cut region of the virtual slide. The pathologist could then add the cross description of the biopsy specimen, the findings in the microscopic image, and the corresponding pathology diagnosis in the report. A URL appears in the image tag of the cut region, which is linked to our

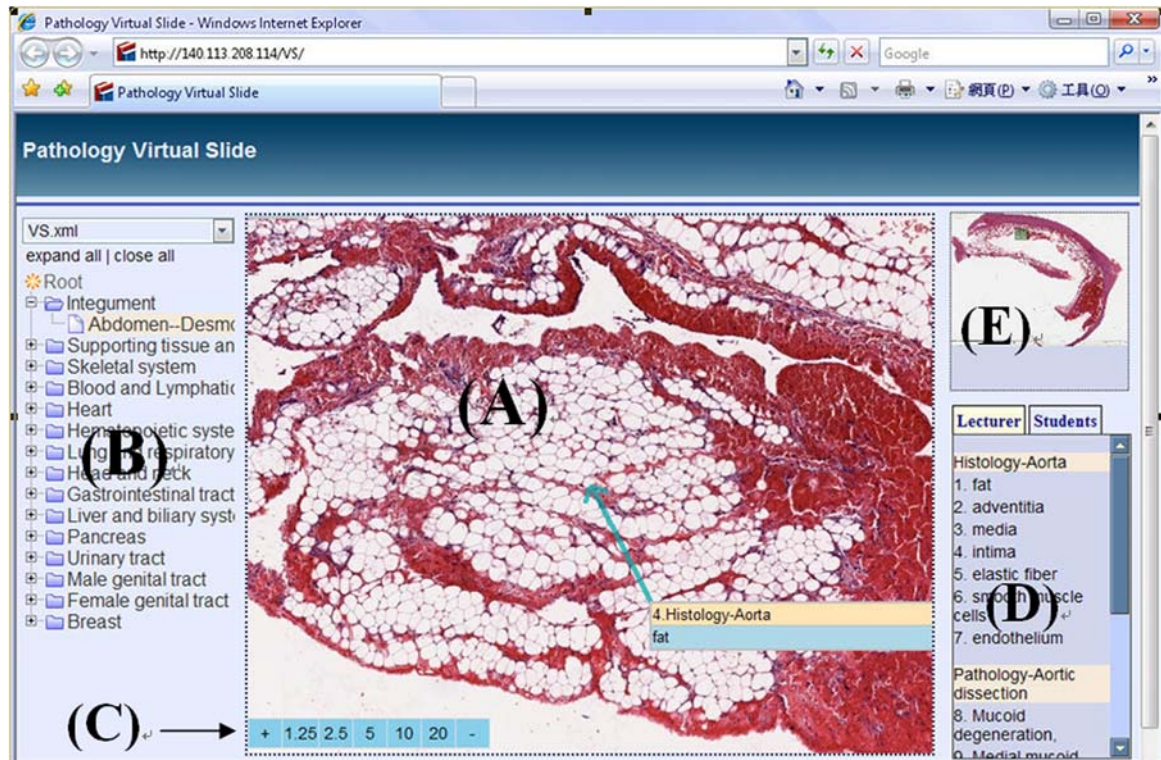


Fig 7. Screenshot of the virtual system. A The central canvas displaying the virtual microscope image and annotations by mouse click and drag. B The slides menu with a tree structure for jumping the different slide. C The zoom function changes the resolution level by mouse click. D The annotation system. E The overview image with a marker indicating the current view position (*translucent rectangular area*).

virtual slide viewing system. Clicking on the image will open the original virtual slide web page, so the pathologist could easily access and navigate the original virtual slide through the link. Because the report is a pure web page, it is easy to modify the format and interface of the report for domain-specific requirements.

DISCUSSION

Web Solutions for Constructing Virtual Slide Systems

Web solutions have been used in many domains to handle multimedia content, and there exist powerful tools and technologies for handling web-based multimedia content. However, when one tries to select web technologies for handling multimedia clinical contents such as the virtual slide images demonstrated in this paper, some

considerations must be addressed. There are many different kinds of web browser provided by distinct providers running in a heterogeneous healthcare environment, and some advanced web technologies for handling multimedia content may not be supported by some currently used browsers. We therefore need a web solution that is supported by most browsers, and the content must be displayed consistently (presentation consistence) in all browsers. Long-term preservation of the virtual slide content is another consideration. Information systems change and evolve very rapidly: In a few years, newer versions of platforms and browsers (for example, Windows 2010 and IE9.0) might be used to view clinical content constructed at the present time. The content therefore needs to also be presentable in new viewing systems. Some proprietary technologies supported by specific browsers might not be appropriate to use for sharing clinical content (for example, using ActiveX controls or Java plug-in

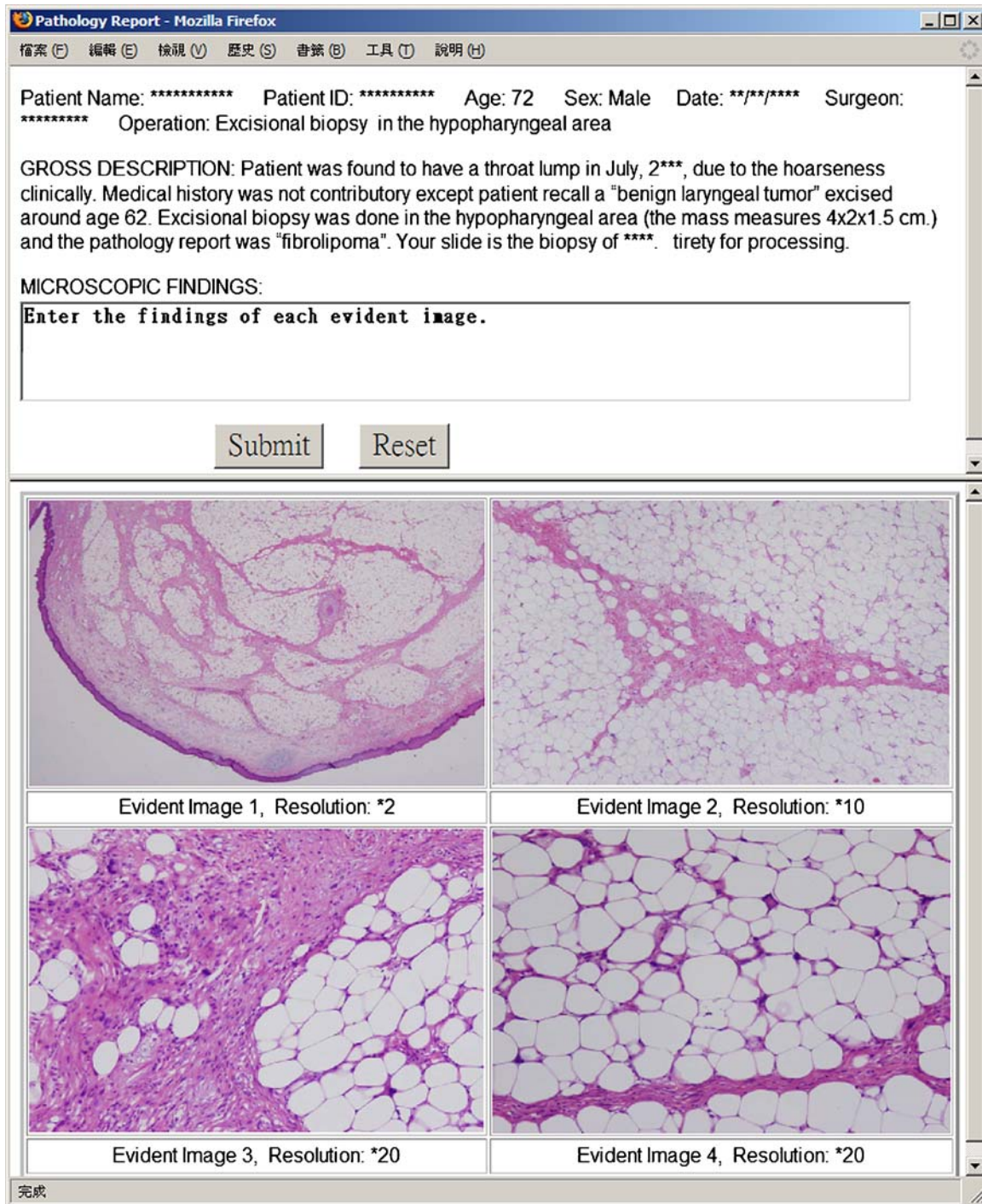


Fig 8. Screenshot of the web-based evidence report.

components) in a heterogeneous healthcare environment. It is known that some popular browsers and platforms have backward version support problems, and this will cause problems each time

platforms and browsers are updated. Plug-in components used in current browsers may be not supported by future upgraded systems and environments; therefore, the selection of widely ac-

cepted technologies that are supported by all web environments is a key factor in the construction of clinical multimedia content.

The HTML- and JavaScript-formatted virtual slide images could be considered as system-independent clinical documents, as HTML and JavaScript are stable and mature technologies supported by all browsers. In this paper, we have demonstrated that this solution is appropriate for handling virtual slide microscopic images. The system has been tested in various browsers (IE6, IE7, Firefox 2, Opera 8, and Safari 2). During the testing process, we made some modifications to eliminate JavaScript codes that are not supported by all browsers. Our virtual slide images were placed on three different web servers (two apache servers and one MS IIS server). On one of the apache servers, we installed a moodle learning system for the pathology course and content management. Two separate groups from different campuses in National Chiao Tung University and NYMU use PHP running in an apache server and ASP.Net running in an IIS server to handle (account and content management) our virtual slide web content. There were no difficulties in manipulating the virtual slide web content in all three web systems. Even without a web server, we can directly open and view web pages on browsers or directly access the images on a local computer, CD, DVD, or flash disk. In this paper, we have shown that this solution is powerful enough to properly handle and present digital microscopic images, and there are no problems for browsers in displaying the pure HTML and JavaScript content or for web servers in managing well-known structuralized data. The pure web solution is suitable for sharing digital microscopic images in the heterogeneous healthcare environment. To the best of our knowledge, such capabilities are still not being fulfilled by many sophisticated technologies and solutions. Our pathology viewing and report system could also be used in a clinical environment. The created digital virtual slides are part of digital clinical documents that should be preserved for years. HTML and JavaScript are well accepted in the Information Technology domain. The presentation of HTML-formatted web pages is the same in all browsers; hence, the presentation consistence requirements of clinical documents can easily be achieved using the pure web solution. As a part of clinical content, the pure

HTML solution is suitable for sharing in the heterogeneous healthcare environment and for long-term preservation when systems and technologies are rapidly changing.

WADO Accessing of Virtual Slide Images

For the purpose of viewing medical images on the web, DICOM has defined an HTTP protocol, Web Accessing to DICOM persistent Objects (WADO), for accessing and viewing DICOM images. WADO uses HTTP requests with parameters to access DICOM objects. The request parameters include UIDs (study instance UID, series instance UID, and SOP instance UID), MINE type, and image presentation parameters. Our web solution for accessing pathologic virtual slide images is similar to the WADO protocol, and it would be appropriate to adopt UID parameters for identifying and accessing virtual slide images in a clinical pathology workflow. However, as described in this paper, some techniques have been created for presenting huge virtual slide images and annotations on the web, and although WADO defines parameters for adjusting image size and adding annotations to images, the specifications defined in the current WADO were designed for radiologic images (small images compared with pathologic virtual slide images). Therefore, the presentation parameters in the current WADO standard might not be appropriate. For example, WADO supports image size and region parameters in request parameters (rows, columns, region [X1, Y1, X2, Y2]) to identify the size of the response region and the area of the whole image plane [0, 0, 1, 1]. Following current WADO specifications, the server should reconstruct the jpeg image corresponding to the specified area in each WADO request. The WADO request would be appropriate for accessing a specific region displayed on the web; however, in a virtual slide viewing system, the viewing region changes each time we adjust the resolution or location of the viewing canvas. If we adopt the WADO protocol in a virtual slide system, the response time of image viewing functionalities running on a browser will be greatly increased. It is more convenient to transmit blocks of images to the browser and use those blocks to construct the region on the browser (as described in the "Method" section). Use of the WADO protocol to access a single region of the whole

pathologic image is supported by our virtual slide system; however, current WADO specifications are not suitable for the construction of a virtual slide viewing system.

CONCLUSIONS

In this paper, we have demonstrated that a widely accepted web solution (using HTML and JavaScript) can handle and present large digital microscopic images on the web. There is no need to use proprietary protocols and plug-in components to construct a web-based virtual slide viewing system. As demonstrated using our viewing system, the functionalities (adjust display resolution, move display canvas, identify navigation location, and show multimedia annotations) can be accomplished using a pure HTML and JavaScript web solution. The solution is easily integrated with other web solutions; it is also easy to adjust the presentation and functionalities of our HTML web page to fit various clinical requirements, as in the example of pathology reports in this paper.

REFERENCES

1. Glatz-Krieger K, Spornitz U, Spatz A, Mihatsch M, Glatz D: Factors to keep in mind when introducing virtual microscopy. *Virchows Arch* 448:248–255, 2006
2. Kumar RK, Velan GM, Korell SO, Kandara M, Dee FR, Wakefield D: Virtual microscopy for learning and assessment in pathology. *J Pathol* 204:613–618, 2004
3. Kumar RK, Freeman B, Velan GM, De Permentier PJ: Integrating histology and histopathology teaching in practical classes using virtual slides. *Anat Rec B New Anat* 289B:128–133, 2006
4. XML-based virtual slide box for teaching histology and pathology, <http://cclcm.ccf.org/vm>; accessed 31 August 2007.
5. Harris T, Leaven T, Heidger P, Kreiter C, Duncan J, Dick F: Comparison of a virtual microscope laboratory to a regular microscope laboratory for teaching histology. *Anat Rec B New Anat* 265:10–14, 2007
6. Helin H, Lundin M, Lundin J, Martikainen P, Tammela T, Helin H, van der Kwast T, Isola J: Web-based virtual microscopy in teaching and standardizing Gleason grading. *Hum Pathol* 36:381–386, 2005
7. MarchHevsky AM, Relan A, Baillie S: Self-instructional “virtual pathology” laboratories using web-based technology enhance medical school teaching of pathology. *Hum Pathol* 34:423–429, 2003
8. Dee FR, Lehman JM, Consoer D, Leaven T, Cohen MB: Implementation of virtual microscope slides in the annual pathobiology of cancer workshop laboratory. *Hum Pathol* 34:430–436, 2003
9. Massone C, Peter Soyer H, Lozzi GP, Di Stefani A, Leinweber B, Gabler G, Asgari M, Boldrini R, Bugatti L, Canzonieri V: Feasibility and diagnostic agreement in tele-dermatopathology using a virtual slide system. *Hum Pathol* 38:546–554, 2007
10. Ho J, Parwani AV, Jukic DM, Yagi Y, Anthony L, Gilbertson JR: Use of whole slide imaging in surgical pathology quality assurance: design and pilot validation studies. *Hum Pathol* 37:322–331, 2006
11. Clarke GM, Zubovits JT, Katic M, Peressotti C, Yaffe MJ: Spatial resolution requirements for acquisition of the virtual screening slide for digital whole-specimen breast histopathology. *Hum Pathol* 38:1764–1771, 2007
12. Glatz-Krieger K, Glatz D, Mihatsch MJ: Virtual slides: high-quality demand, physical limitations, and affordability. *Hum Pathol* 34:968–974, 2003
13. The advantages of line scanning for rapid entire slide capture. <http://www.aperio.com/productservices/prod-technology.asp>; accessed 31 August 2007.
14. Sun C, Beare R, Hilsenstein V, Jackway P: Mosaicing of microscope images with global geometric and radiometric corrections. *J Microsc* 224:158–165, 2006
15. Prototype JavaScript framework. <http://www.prototypejs.org>; accessed 31 August 2007.