International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(B)**, October **2012**

 ISSN 1349-4198
pp. 7225–7235

IMPLEMENTING A ZOOMABLE WEB BROWSER WITH ANNOTATION FEATURES FOR MANAGING LIBRARIES OF HIGH QUALITY IMAGES

Dimitris Kanellopoulos[1], Petros Lalos[2] and George Tombras[2]

[1]Department of Mathematics
University of Patras
Patras 265 00, Greece
d_kan2006@yahoo.gr

[2]Department of Electronics, Computers, Telecommunications and Control
Faculty of Physics
National and Kapodistrian University of Athens
Athens 157 84, Greece
{ plalos; gtombras }@phys.uoa.gr

Received July 2011; revised February 2012

Abstract. *Viewing high-resolution images on the Web is a common activity. For this purpose, specialized zoomable browsers have been proposed to reduce the network traffic. This paper presents a lightweight and platform independent zoomable browser that supports annotations of the high quality digital content. The browser can display high-resolution images in resource-limited mobile environments like PDAs, PALMs and smartphones, even in low speed connections. We evaluated the browser and the main finding is that data transfer is reduced with the aid of the browser, while the presented information is of high quality and could be used in various operating environments. The browser is developed using lightweight Web technologies, which makes the tool ideal for downloading high detailed digital content.*
Keywords: Media content management, Image slicing, High-resolution images, On-line annotation, On-line collaboration, Image browser, Multimedia

1. **Introduction.** High-resolution images are used in culture, astronomy, engineering, architecture, publishing and e-commerce. For example, many civil structures need periodic examination for proper operation and/or for maximizing life span. For structural safety monitoring, a critical element is structural damage diagnosis using high quality images. Fu and Moosa [1] presented a new approach to diagnosing structural damage using high-resolution images, which can be provided by a charge coupled device (CCD) or CCD camera. In addition, high-resolution images can be used in the process of land use dynamic monitoring and object recognition [2]. The advance of collections of high-resolution images requires proper image-browsing tools. The need to easily and quickly view image data over the Internet has motivated researchers to develop various image transmission schemes [3]. Such schemes are designed to reduce image file sizes and maximize the speed at which a user views remotely-located images. In general, Internet users view high-resolution images using various Web applications. In this case, one problem arising is that clients have to download a large amount of information for high quality images, usually through limited bandwidth connections. This process is time-consuming. This problem is even more noticeable when a user wants to navigate in high quality image collections. For the publisher of the digital content, the network load constitutes a major bandwidth and server availability problem, as it maximizes the effort of all the involved

network resources. Viewing digital content in devices with screen analysis lower than the analysis of the image can cause additional display problems. However, users must be able to view the content with handheld devices like PDAs and smartphones without having to compromise the offered features. To solve the above problems, we created a generic zoomable image browser. The browser displays detailed multimedia content like images, or other digital material that could be transformed to images (e.g., PDF files) in an interactive, manageable and user-friendly way, with minimum network and server demands. Users can zoom into specific areas of the images to see high-resolution illustrations of these parts. Another feature of the browser is the placement of annotations on the portrayed content to support collaboration between viewers of the digital content. Let us present an example that shows how the scientific community could benefit from the use of the proposed browser. Three life science laboratories are collaborating on a combined cell experiment with the participation of many specialized researchers. Imaging live cells is a complex task even for experienced microscopists. Because many research questions can only be addressed in living cells (like cellular dynamics), they require live observation and collaboration over time. Each laboratory acquires an image data set with the use of electronic microscopes. They share it with their colleagues at the different sites via the web-based image data management system of the proposed browser. Collaborating life-scientists can monitor a cell from different perspective viewports via high-resolution images, and examine a cell event from all angles. Life-scientists can also collaborate in real time and simultaneously annotate specific interest points of the shared images.

2. **Related Work.**

Many organizations publish their online digital content in multiple files formatted as PDF or JPEG. Users who need to view selected parts of these files have to download the complete file. This means that users need to download large numbers of data in order to view the desired segment of the content. Some solutions for alleviating these problems are based on partitioning the content in smaller segments [4, 5, 6]. The user can interact and choose a specific segment of a page and view it as an image in a separate web page. In this way, only a segment of the information is delivered in detail, and this reduces the network load to only the fraction of the delivered segment. The basic process of image segmentation is edge detection. Well-known methods of edge detection are the Sobel filter, and Derivative of Gaussian and Laplace of Gaussian. However, these methods cannot detect edge details and it remains difficult to segment thin areas because the edge pixels obliterate details. To solve this problem, an image segmentation method has been proposed by using boundary code. ZoneZoom lets users traverse large information spaces on smartphones [7]. This tool segments a given view of an information space into 9 subsegments, each of which is mapped to a key on the number keypad of the smartphone. ZoneZoom allows large images to be viewed in small screen devices. Instead of having a single page visible at a time, multiple pages and the links between them are depicted on a large zoomable information surface [8]. Links are shown as a hierarchy so that user can see the relationship of web pages with their parent and child nodes. Desktop PhotoMesa is a zoomable image browser for desktops [9]. It enables users to view hierarchies of directories in a treemap layout and browse through them in an animated navigation mechanism. PhotoMesa uses quantum strip treemaps to arrange images over a single screen. It also uses semantic zooming to save different sets of thumbnails for each zooming level. Pocket PhotoMesa is an adaptation of the original implementation for PDAs [10]. Another browsing system for fast link traversal on large screens is presented in [11]. Here, the system helps users decide which links to follow by providing zoomable thumbnails of the pages pointed to by the hyperlinks. In another solution [12], digital files are divided into smaller segments, but they are utilized in a different manner. The viewer interactively chooses each segment, which is delivered in a vendor predefined way in order to minimize the network load. This means that if the chosen segment contains text, it is presented in a web-oriented way (i.e., simple HTML format) and is published in a specific spot of the delivered web page. This method is quite common and is utilized in many popular image browsers. In [13], a better temporal organization of the zoomable content is presented. In [14], the authors present the design and evaluation of a simple interface technique that allows rapid document navigation. Abowd et al. [15] explore the use of a zooming interaction paradigm for browsing and filtering large collections of video scenes. An early evaluation of zoomable tools can be found at [16]. Cockburn et al. [17] review interface schemes that allow users to work at, and move between, focused and contextual views of a dataset. They review and categorize these schemes according to the interface mechanisms used to separate and blend views. Cockburn et al. [17] identified four main approaches:

- overview+detail which uses a spatial separation between focused and contextual views;
- zooming which uses a temporal separation;
- focus+context which minimizes the seam between views by displaying the focus within the context; and
- cue-based techniques which selectively highlight or suppress items within the information space.

However, none of these approaches/tools is ideal. For example, it is reported that the abrupt transitions between discrete zooming levels meant that the participants had to reorient themselves with each zoom action [18]. The main conclusion is that several issues are still open, and so more systems and approaches to present complex information to the users are needed. Most of the proposed solutions divide data into segments and deliver the content, based on a user specific request. The delivered data is just a fragment of the whole information and the content is displayed in detail only when the viewer requests it. However, existing solutions have some disadvantages. Their browsing method is not generic as these solutions are used for specific cases or platforms. Clients also have to download specific applications in order to operate as in the case of GoogleEarth, which can cause problems in limited resource environments. The proposed browser is based on similar image slicing principles, but it is developed as a lightweight Web application in Adobe Flash. Based on the implementation technology, the proposed browser can be utilized in limited resource environments. The browser offers a variety of advanced image processing and annotation tools to accurately analyze, import, annotate and export precision digital images. It opens up a world of possibilities for life scientists, researchers, engineers, students, astronomers, and other professionals who regularly gather, analyze data from scientific imagers, scanners and custom data collection instruments and need to share their findings and collaborate on them. Application areas where the browser offers added value include: (1) education and entertainment; (2) publishing agencies; (3) museum, galleries and image archives; (4) wireless and handhelds; (5) mapping, GIS, real estate and travel; (6) aerial photos and space imaging; (7) medical and scientific; (8) defence, security and infrastructure; (9) industrial quality assurance; (10) materials science and metallurgy.

3. **The Generic Zoomable Browser.** The key-idea behind the browser's design is to have a flash application embedded in a web page running on the client side and that has most of its features controlled by external variables. These variables are handled by the client with the use of an external dynamically generated XML file. The XML file contains the details regarding the images location, content metadata, and personalization performance specifications, which are user-defined. These details can be re-adjusted at

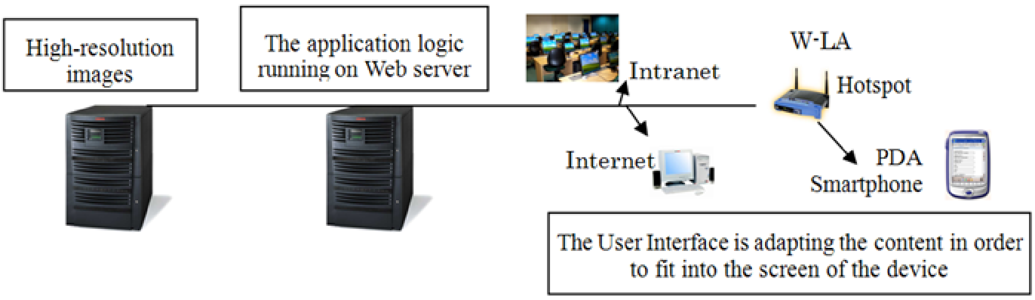

FIGURE 1. High-level architectural view of the system

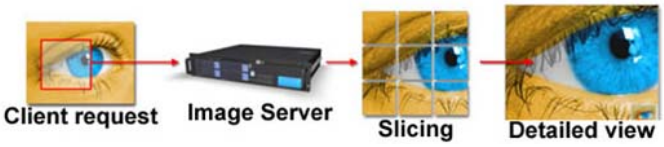

FIGURE 2. Slicing a high-resolution "mother image" in pre-defined image tiles

anytime during the utilization of the tool. These performance details are mainly about the focus and zoom in or out percentage, which finally defines the analysis of the projected image. The application server delivers the sliced images in the optimized resolution, based on the image and focusing requests of the client. With the use of a script language, the application generates a data and variables map in the form of a personalized XML file. The information is transformed and published by the browser with the combination of an XSLT (Extensible Stylesheet Language Transformations) file, depending on the platform (device heading, i.e., PC, PDA or smartphone) used by the client (Figure 1). Every aspect of the browser's zoom-in technology has been optimized: (1) processing is distributed; (2) downloads of image data are minimized; (3) requests of redundant data are prevented; (4) IT costs are minimized; and (5) the end-user experience is maximized.

The workflow of the presented application can be summarized in the following steps:

(1) An online user views a low detail preview of the published content. This preview image is automatically extracted by the image server and delivered to the clients' browser environment.
(2) Clients interact with the browser and choose a specific part of the preview image they would like to focus on.
(3) Then, this request is delivered to the application, which "slices off" a higher resolution image of the specific requested part of the content. The application delivers it to the client through the browser's interface.
(4) If the viewers need to go into higher detail, they request a more detailed result. The image server slices off a segment of the previous part, but in higher resolution from the initial image, and delivers it to the requesting viewer. This mechanism describes the fundamental steps of the image slicing procedure (Figure 2).

The images are pre-processed to create the specific slices. New images can be added on the fly and the image server can slice them on demand. The procedure starts with a high-resolution image of the original image. The settings of the application were adjusted to create hierarchically structured sub-layers of halved resolution content in each lower layer. The produced lower resolution copies of the image depend on the initial image.

3.1. The zooming process.

- The provider uploads one or more high-resolution images ("mother images") that will serve as the foundation for lower resolution images. These mother images are indexed, mapped and stored into the data server of the application under the name of the specific viewer.
- Users download the browser along with thumbnails of the existing images by clicking on the appropriate URL. Then, they focus on a specific area using the various focusing features provided by the browser (mouse wheel, click on or button). Each of the viewers' requests is "translated" by the browser into URL requests that are delivered to the Application Server to run the appropriate Server side scripts. The data server delivers the proper resolution of the image based on the focus intensity (as requested by the Application Server). If the focus intensity is maximized, it uses the "mother image" and the proper sliced tiles.
- If the focus intensity is lower, then it uses a lower resolution image and changes between the available resolution levels.
- The browser generates and provides the image server with detailed coordinates of the requested area. Then, the image server slices the specific tile/s corresponding to the coordinates and serves the resulting image tiles back to the browser.

To increase the browser's efficiency, the use of all possible caches both server-side and client-side has been included. The browser's solution is entirely standards-based, and is therefore perfectly suited to leveraging existing efficiencies of web servers, operating systems, browsers, and the Flash Player. Any requested image 'tile' at any resolution is simply a JPEG file and will thus be automatically managed according to the web server cache settings. Image tile part requests are fulfilling byte-range requests, a standard HTTP 1.1 protocol function just like fulfilling file requests, which is what Web servers are designed for. These tiny JPEGs are stored on the client machine by the browser for fast subsequent accesses by the Flash Player. All logic and processing occur on the client. As the user views an image via the browser, it is his system determining what the current view is, and what image data is needed for that view. The browser delivers maximum efficiency for maximum performance. For example (a worst-case scenario given hardware and bandwidth available at reasonable cost), on a dual core Pentium 2,5 GHz with 2 GB of RAM memory assuming throughput of at least 2 saturated 100 mbps Fast Ethernet connections, theoretical throughput is 2×100 mbps = 200 mbps. Effective real throughput (40%) is 80 mbps = 10,000 KB/s. At 9.5 KB per 256-pixel tile this represents 1053 tiles per second. At approximately 2.5 tiles per second per user (that is, a half screen view every 2 seconds) this supports 421 concurrent users with a 1 second response time (this assumes 125 concurrent users with only one Ethernet connection).

Viewing images with the browser prevents all unnecessary downloads. Image data is requested by the browser only for the parts of the image that the user actually looks at and only at the zoom level they actually zoom-in to. This provides a significant bandwidth economy as compared to common implementations involving large JPEGs, where the entire image is downloaded even if two thirds of the image will not fit on the user's screen and only the center third is ever viewed. Two factors are defining 'image zooming'. The first is the zoom-in percentage and the second is the projected final outcome of the view. The zoom-in percentage (meaning how much you are able to zoom-in) depends on the size of the source image as compared to the size of the intended display area. An image that is 1280×1024 pixels displayed on a 1280×1024 monitor, when zoomed-in will offer no zoom at all. The image will just fill the display area if zoomed-in at 100%, although it is possible to go beyond 100% but then the stretching will be evident through pixelization.

For this reason, a source image should best be 2000 to 3000 pixels in width and/or height at a minimum.

The final view after zooming is also important because it determines whether the zooming procedure is useful and satisfying. For example, scanning a 48-bit 2400 dpi of a 35mm negative image of a crowd ends up with around 20 megabytes of data. The initial image with thousands of pixels in each dimension provides much more detail than is needed for most users. When zooming in on the image of the crowd the user expects to be able to clearly see a face. A user with high demand on detail will expect to clearly depict detail (e.g., see blood vessels in the eye of the face).

3.2. **The browser's environment.** Users can utilize the system by downloading a version of the browser and run it through their Web browser. Then, they can start navigating through the available images. Figure 3 shows the administration page. The users can identify the specific page number of a thumb by pointing their mouse over the thumbnail and

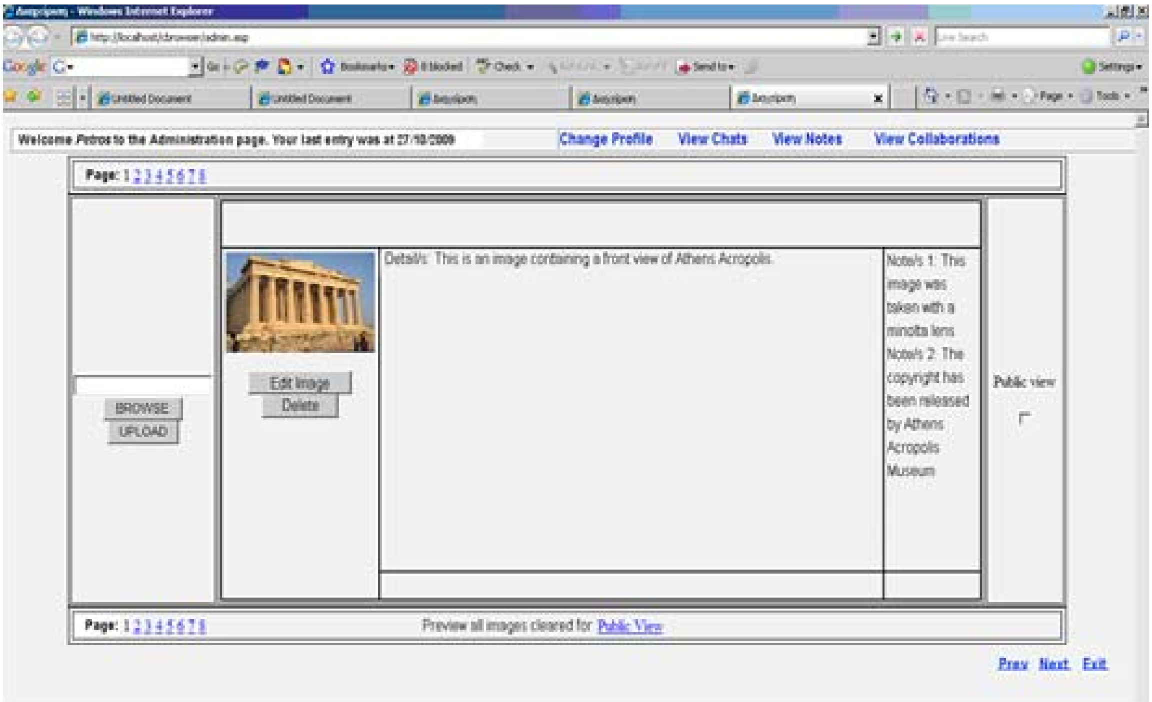

FIGURE 3. The administration environment

FIGURE 4. The main stages of the browser with the thumbnail slide show

```
<?xml version="1.0" encoding=iso-8859-7"?>
<viewer>
<engine></engine>
<thumb>thumbs</thumb>
<image imgid="1" page="1/2" pageA="1" pageB="2"
WIDTH="3276" HEIGHT="2042"
PHOTO="http://10.1.1.28/img1.jpg" XPos="0" YPos="0"
ZLevel="23" MaxZoom="200" MinZoom="23" zoomfactor="1.3"
deltaFactor="0.8" />
...
```


Figure 5. An example of XML file that contains details (i.e., first image focus intensity and position, response degree to the visitors focus request)

reading some metadata about the page as a tooltip. Alternatively, they may also choose to use the search page for a specific keyword search via the metadata repository. The users' different views are presented by the following figures (Figure 4).

The application was created with the use of Adobe Flash and ActionScript. The browser locates the images directories by using an external XML file which the script language generates dynamically. The XML file contains details about where the pages (in the form of JPEG images), called the dataset, are stored (Figure 5). In this example, variables like XPos, YPos control the starting coordinates of the published page. ZoomFactor controls the intensity of the focusing mechanism, while deltaFactor controls the immediacy of a viewer's mouse wheel reaction. The Script Language at the Business tier generates the new addresses and supplies XML feedback for the browser.

3.3. **Annotation and collaboration features.** Two key advantages of inserting annotation into a high-resolution image are the ability to share these notes with other users and being able to access the annotations from any Web enabled computer. Russell et al. [19] developed a Web-based tool called "LabelMe" that allows easy image annotation and instant sharing of such annotations. The proposed browser provides an annotation tool that allows viewers to place one or more annotations on specific parts of the published images. Viewers can place annotations on an image and store them for later reviewing (Figure 6(a)) or use the chat tool that has been implemented for synchronous communication (Figure 6(b)). Online chats are being stored and can be recalled anytime by the cooperating chatters. Viewers can use this tool to collaborate online and maximize the coactions' outcome. The browser provides annotation and collaboration features in the form of simple software tools, allowing real-time collaboration among interested parties in the context of web-shared images of common interest. A user can annotate specific points of the zoomed-in image and share comments among collaborators or exchange views.

4. **Evaluation of the Zoomable Browser.** The first aim of the evaluation was to understand whether there is a saving in the transmitted data or not in using the proposed tool. Secondly, we wanted to record the opinions of the users with respect to the usefulness of the supported features.

4.1. **Method and data.** We first created a list of 200 high quality images of different domains. All the images where in jpeg format, downloaded from the Web and were preprocessed in order to be accessible through the zoomable browser. The size of each image was greater than 1MB and its resolution higher than 1600×1200 pixels. Then, 20 Web pages containing the thumbnails of 10 of available images were created. Next to each thumbnail, two links were created pointing to the original high quality version of the image and to the zoomable browser's version. Both links were similar and randomly put

(a) (b)

Figure 6. Posted annotation attached to the specific spot of the image independent of the zoom grade requested & annotation and collaboration features (online chat)

in the page, in order to prevent users from systematically selecting one over the other. The users of our study will then use these Web pages and the links in order to access the high quality content. All the files were in the same server. The 25 subjects of the evaluation study were University students. All the users were first year undergraduate students, with advanced computer handling abilities. They used the Web and some social networks on a regular basis and were quite familiar with image exchange and upload/download in the Web. Each of the students was asked to access one of the Web pages and to view the images through a PC with a fast Internet connection. It was explained to them that each link depicts the same image but with a different mechanism. Since students accessed different pages and the images were depicted using a different mechanism no caching of images was possible which could affect the experiment. Students accessed the image server simultaneously from different clients located in different networks in order to create increased network demands and requests to the server. In the second experiment, the subjects were sequentially provided with the same palmtop and were asked to access a web page containing 10 images, which were not used in the previous experiment. The students could access the same image with the regular way and the zooming facility. The participants were asked to visit at least three images by selecting both of the ways. After these two experiments, the users were asked to fill a short questionnaire and to share with us their experiences in using the zoomable browser.

4.2. **Results.**

4.2.1. *First experiment.* During the image navigation process through the PC, we recorded the data downloaded by the users in order to understand the savings in data transferring. At the end of the evaluation it was found that 132 images viewed in both ways; 9 images only with the zoomable browser and 7 images only in the usual way. In 16 cases, the downloading of the regular image was abandoned during the downloading by user the due to the increased required time. This is an indication that accessing high quality images in sequence in the Web, is a procedure requiring fast mechanisms to efficiently support the users. Table 1 shows the size of 10 high quality images and the size of the downloaded data when the images were viewed through the zoomable browser. We selected the five best and the five worst cases. As seen the saving in data transfer varies from approximately 19% to 87%. On average, the saving in download was approximately 42%, which means an actual saving of over 60 MBs of data in this experiment. The saving in data depends

TABLE 1. Reduction in transmitted data

Size of the high quality image	Downloaded kilobytes through the zoomable browser	Reduction %
1541 KB	187 KB	87.87%
1334 KB	234 KB	82.46%
1341 KB	322 KB	75.99%
2340 KB	568 KB	75.73%
1222 KB	387 KB	68.33%
1909 KB	931 KB	51.23%
1456 KB	908 KB	37.64%
1802 KB	1201 KB	33.35%
1900 KB	1345 KB	29.21%
1805 KB	1459 KB	19.17%

largely on the user actions, meaning that if the users focus on the central theme then they will have a large saving. Otherwise, they saving in data will be smaller but even in this case it is considerable (Table 1). Another issue was the measurement of time spent to view the images in both cases. We measured the time from the moment a user requested an image to the moment they left the page depicting the image and returned on the list of thumbnails. In all the cases the time spent in viewing the images in the regular way was increased compared to the time used to see the image through the zoomable browser. On average each user spent approximately 14 seconds viewing the image through our browser and 23 seconds in the usual way. These figures include both the time needed in downloading the image and the time spent in viewing it, as it is difficult due to the gradual depiction of the image through the browser. However, it is clear that the users spent less time on each image with the zoomable browser.

4.2.2. *Second experiment.* The users were asked to access less images due to the limited available hardware. In this stage we measured (again) the percentage of data reduction and the time spent in downloading and viewing the images. In this case, the saving varied from approximately 49% to 79% with an average reduction of 68%. Most users viewed a medium sized version of the available images through our browser. The time spent in downloading and viewing the images was again reduced, with an average of 12 seconds through the zoomable browser and 28 seconds when downloading the high resolution version of the image. Both experiments show that the savings in downloaded data and time, are increased in our web browser. Especially in the limited resource environment, these reductions were more apparent. These data are strong indications of the value of the tool for viewing high quality content.

4.2.3. *Questionnaire.* A questionnaire was distributed to the participants of the evaluation. The aim of the questionnaire was to record the overall satisfaction of the browser's users in terms of easiness and manageability, in both environments. The questionnaire consisted of 14 questions scored in a three-point Likert scale. Table 2 presents the questions set to the participants and their replies.

A discussion with each participant was performed at the end of the questionnaire, to record their specific views. The first 5 questions concerned the easiness of using the environment and the help system. As seen, learners could easily learn and use the tool and no technical difficulty was observed or mentioned by the users. The answers are divided in the fourth question because as it was explained by some of the participants, although the tool was easy to use, they had to zoom in and out, which makes the process

Table 2. Questions asked to the participants

	Questions	I agree		Undecided		I disagree	
		N	%	N	%	N	%
1	Using the browser was easy	19	76.00%	6	24.00%	0	0.00%
2	Learning to use the browser is a demanding task	0	0.00%	6	24.00%	19	76.00%
3	I had problems in using the browser	0	0.00%	3	12.00%	22	88.00%
4	I enjoyed using the browser	15	60.00%	3	12.00%	7	28.00%
5	The help system was useful	3	12.00%	22	88.00%	0	0.00%
6	By using the browser, it was easier to understand the main theme of the image in the PC	25	100.0%	0	0.00%	0	0.00%
7	Viewing an image with the browser was faster in the PC	25	100.0%	0	0.00%	0	0.00%
9	Viewing multiple images with the browser is easier in the PC	25	100.0%	0	0.00%	0	0.00%
10	I prefer the usual way for viewing images in the PC	11	44.00%	9	36.00%	5	20.00%
11	Using the browser, it was easier to understand the main theme of the image in the mobile device	25	100.0%	0	0.00%	0	0.00%
12	Viewing an image with the browser was faster in the mobile device	25	100.0%	0	0.00%	0	0.00%
13	Viewing multiple images with the browser is easier in the mobile device	25	100.0%	0	0.00%	0	0.00%
14	I prefer the usual way for viewing images in the mobile device	0	0.00%	5	20.00%	20	80.00%

more demanding. Questions 6 to 10 concerned the efficiency of using the browser in the desktop environment. As seen, all users agreed that it was easier to understand the main theme of the image. Further, they argued that it was faster to view a single or multiple images. These responses are indications that the main advantage of the tool is indeed the efficient in downloading and viewing high quality images. The responses in question 10 were more distributed and mostly in favor of the common method for viewing images from the Web. In the discussion after the questionnaire, the participants argued that although they realized that it is by far faster to view images through our image viewer, it is easier to just click and download images. In our browser they need to zoom in on to the selected areas to view them in more detail, which makes the system's utilization more demanding in terms of interaction. The last set of questions concerned the utilization of the browser in the mobile device. Again, the responses were positive towards our tool, as the evaluators argued that the tool allows them to view images faster. Further, in this case they prefer to use the zoomable browser instead of the usual way, as it is actually easier in the specific environment.

5. **Conclusions.** We introduced a browser with zooming facilities that allows users to view high quality images faster than downloading them through the Web. The evaluation showed an average saving of 42% in the downloaded data in the desktop environment and 68% in the mobile device. This enables the browser ideal for browsing libraries of high-quality images. Further, users have an immediate view of a low-resolution instance of the image and can decide instantly whether they want to view a more detailed version of it. The browser is a lightweight application with minimum hardware requirements, and can run on limited resource environments. Another point of innovation is the utilization of XML for storing the optimization details and the other data needed for performing the zooming in or out. By logging into the tool, it is possible to search for images and through the other available options to have an integrated environment for fast access to high quality content. An additional function of this browser is that users can place annotations on the images and also chat on these comments in a synchronous mode. This makes the tool ideal for collaboration among various categories of users who need to work

and share high resolution images in order to complete their tasks. The proposed browser operates efficiently over low bandwidth network connections or when increased amounts of data need to be downloaded.

REFERENCES

[1] G. Fu and A. Moosa, Structural damage diagnosis using high-resolution images, *Structural Safety*, vol.23, no.4, pp.281-295, 2001.

[2] X. Zhu, F. Mao and J. Huang, Land use dynamic monitoring using multi-temporal SPOT data in Beijing city from 1986 to 2004, *IEEE Int. Geoscience and Remote Sensing Symp.*, vol.4, pp.IV-699-IV-702, 2008.

[3] T. Liptay, J. Barrou and I. Gargantini, A WWW interactive progressive local image transmission system, *Internet Imaging Conf.*, San Jose, CA, USA, vol.3964, pp.330-343, 1999.

[4] L. Good, Staying in the flow with zoomable user interfaces, *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, Minneapolis, MN, USA, pp.548-549, 2002.

[5] L. Good and B. Bederson, Zoomable user interfaces as a medium for slide show presentations, *Information Visualization*, vol.1, no.1, pp.35-49, 2002.

[6] T. Itoh, Y. Yamaguchi, Y. Ikehata and Y. Kajinaga, Hierarchical data visualization using a fast rectangle-packing algorithm, *IEEE Trans. on Visualization and Computer Graphics*, vol.10, no.3, pp.302-313, 2004.

[7] D. Robbins, E. Cutrell, R. Sarin and E. Horvitz, ZoneZoom: Map navigation for smartphones with recursive view segmentation, *Proc. of the ACM Working Conf. on Advanced Visual Interfaces, ACM*, Italy, pp.231-234, 2004.

[8] S. Khawaja, A. Shah and K. Khowaja. Alternate paradigm for navigating the WWW through zoomable user interface, *Advances and Innovations in Systems, Computing Sciences and Software Engineering*, pp.417-420, 2007.

[9] B. Bederson, Photomesa: A zoomable image browser using quantum treemaps and bubblemaps, *Proc. of the 14th Annual ACM Symposium on User Interface Software and Technology*, New York, USA, pp.71-80, 2001.

[10] A. Khella and B. Bederson, Pocket PhotoMesa: A zoomable image browser for PDAs, *Proc. of the 3rd Int. Conf. on Mobile and Ubiquitous Multimedia*, pp.19-24, 2004.

[11] B. Yoo, J. Lea and Y. Kim, The seamless browser: Enhancing the speed of web browsing by zooming and preview thumbnails, *The 17th International World Wide Web Conference*, Beijing, China, pp.1019-1020, 2008.

[12] B. Johnson, B. Shneiderman and B. Treemaps, A SpaceFilling approach to the visualization of hierarchical information structures, *Proc. of IEEE Visualization'91*, San Diego, USA, pp.284-291, 1991.

[13] D. Huynh, S. Drucker, P. Baudisch and C. Wong, Time quilt: Scaling up zoomable photo browsers for large, unstructured photo collections, *Conference on Human Factors in Computing Systems*, pp.1937-1940, 2005.

[14] A. Cockburn, C. Gutwin and J. Alexander, Faster document navigation with space-filling thumbnails, *Conference on Human Factors in Computing Systems*, pp.1-10, 2006.

[15] G. D. Abowd, M. Gauger and A. Lachenmann, The family video archive: An annotation and browsing environment for home movies, *Proc. of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, New York, USA, pp.1-8, 2003.

[16] K. Hornbaek, B. Bederson and C. Plaisant, Navigation patterns and usability of zoomable user interfaces with and without an overview, *ACM Transactions on Computer-Human Interaction*, vol.9, no.4, pp.362-389, 2002.

[17] A. Cockburn, A. Karlson and B. Bederson, A review of overview+detail, zooming, and focus+context interfaces, *ACM Computing Surveys*, vol.41, no.1, 2008.

[18] A. Cockburn and J. Savage, Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods, *Proc. of the British Computer Society Conference on Human-Computer Interaction, People and Computers XVII*, pp.87-102, 2003.

[19] B. Russell, A. Torralba, K. Murphy and W. Freeman, LabelMe: A database and Web-based tool for image annotation, *International Journal of Computer Vision*, vol.77, no.1-3, pp.157-173, 2008.