Creating 3D Web-Based Viewing Services for DICOM Images*

Adam Piorkowski, Lukasz Jajesnica, and Kamil Szostek

Department of Geoinfomatics and Applied Computer Science,
AGH University of Science and Technology, Cracow, Poland
pioro@agh.edu.pl
http://www.agh.edu.pl

Abstract. This paper presents the design of Web-based medical image viewing system. It allows easy access to the selected DICOM data. The purpose of this application is to provide 3D results of computer tomography as a projection at any angle. This system contains of modules for data storing, projection creating and image file creating. In the experimental part two common web technologies (Java and ASP) were tested in terms of performance for designed system.

1 Introduction

The telemedicine and medical image processing has recently become more and more popular. Its main objective is to provide the results of medical examinations on-line, in the real time. Such applications require the creation of special programs that can share, process and transmit a lot of medical data. A simpler solution is to create a web based application that meets the above requirements. This type of program requires only a web browser on the client side. Disadvantages of this case are requirements for performance of hardware (high performance of servers) and performance of software (server-side computing). This article also shows a review for performance of server-side image processing (Java and ASP).

On-demand server-side image processing was the focus of the work by T. Sakusabe et al. in 2000 [1]. The authors presented their own concept of a system that takes advantage of the user's interaction in script languages and processes images on the server's side. They selected and tested the following technologies of applications: ISAPI (currently obsolete, replaced by ASP) and CGI. The method of compression images (PNG vs. JPG) was also tested. The main goal of this work was the performance test for various network connections, bandwidths and client loads. Our considerations relate in some parts to their work.

A new approach to create and manage image-based electronics patient records from actual patient records was a topic of work by Z. Jianguo et al. [2]. There

* This work was financed by the AGH – University of Science and Technology, Faculty of Geology, Geophysics and Environmental Protection as a part of statutory project number 11.11.140.561.

A. Kwiecień, P. Gaj, and P. Stera (Eds.): CN 2009, CCIS 39, pp. 218–224, 2009.

is also presented a way to use Web technology and DICOM standard (Digital Imaging and Communications in Medicine) to build an open architecture for collaborative medical applications. The focus is on transmission security of patients' data in the Internet/intranet.

Paul Ancuta presented a sample 3D modeling of human bone in the paper [3]. To make this visualization technique particularly useful, the author considered a isosurface method. The image is created from successive computer tomography (CT) images. The processing is done by standalone software here.

A multidimensional image navigation and display software was a topic of paper by Antoine Rosset et al. [4]. The authors presented their own standalone solution to display and interpret large sets of multidimensional and multimodality images such as combined PET-CT studies. To optimize display they used 3D graphic capabilities of the OpenGL graphic standard.

Joseph Fernandez-Bayo et al. showed an application for distributing medical images on the Internet [5]. They implemented a server (or cluster of servers) for picture archiving and communication systems (PACS) that served sets of DICOM images over network (under control of Windows NT). They also implemented image viewer (using Java) on the client side. The classic PACS system, which is an open source, was also presented in this article [6].

2 DICOM Images

The results of some medical methods of examination are sets of images. This is the case of e.g. computer tomography. The CT machine creates a cross-section in the form of slices with a given distance away. As a result we get a set of images, which are stored in a special lossless format – DICOM. Although most of DICOM images use grayscale palette, the resolution can be higher (e.g. 12 or 16 bits) than standard display systems (8 bits for CRT, LCD). It is therefore necessary to convert the data to view the images properly.

3 Creating a Projection of 3D Image

To create a projection of 3D image a block of data is created (Fig. 1). Each next slide of CT result is taken as the subsequent surface for Z value. Values (X,Y) are the pixels of surface. The client can ask a server for any slice in block. It can

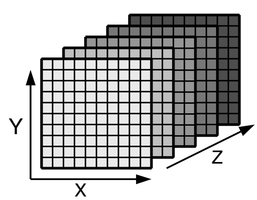

Fig. 1. A block of DICOM images

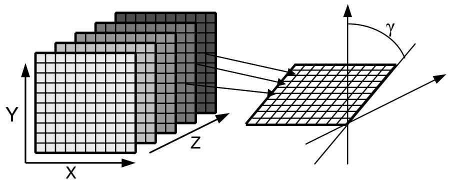

Fig. 2. Creating a projection

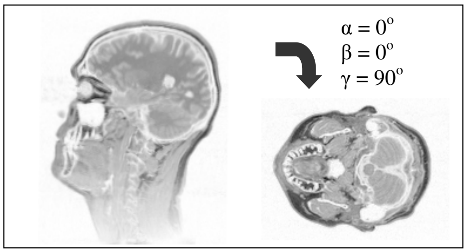

Fig. 3. A sample real projection for medical image

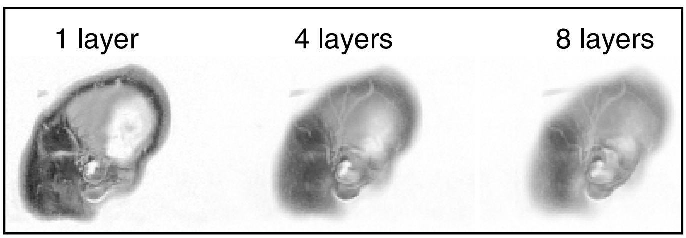

Fig. 4. A 3D view for an ear

also be asked for projection at any angle. To enable any rotation, client should give the parameters – angles:

- $\alpha(x, y)$,
- $\beta(x, z)$,
- $\gamma(y, z)$.

To create that projection for each pixel of this projection we should calculate its new coordinates (after rotation). The sample case is shown on Fig. 2 and Fig. 3 (sample images by: [8]).

To calculate new coordinates we use standard transformation for rotation:

$$\begin{aligned} x' &= x * \cos(\alpha) - y * \sin(\alpha) \\ y' &= x * \sin(\alpha) + y * \cos(\alpha) \end{aligned} \quad . \tag{1}$$

This transformation we should process for three angles (α, β, γ), given by client as parameters of http request. To make a 3D view the program needs to calculate a few of projections (5–10) and join them (Fig. 4).

3.1 Interpolation

The new coordinates are floating point values, but the block of data has indexes as integer values. To get a clear image we should interpolate the value of each pixel. One of the available methods to approximate the value of an intermediate point within the local axial rectangular prism linearly is trilinear interpolation (Fig. 5). The consultants endorsed the quality of this method is satisfactory.

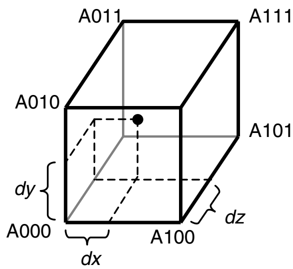

Fig. 5. New coordinates of a point after a rotation

3.2 Server-Side Creating and Processing an Image

At the beginning, the application loads selected CT results to a block of data and next can accept the http requests. The accepting requests consists of the following steps:

- handling a request by web server,
- creating a projection,
- creating an image file from projection (bmp or jpg format),
- posting image file to client.

Processing requests can be done simultaneously.

4 Performance Tests

To determine the response time and throughput for common Web technologies we made tests. There were applications written for Java servlets and ASP technology, running on the web servers (Apache 2.2.9 and IIS 6.0) under control of Linux 2.6 and Windows 2003 Server SP2. We measured the following parameters:

- time of projection processing (in the code),
- time of image file creating (in the code),

– minimal response time for a single request – using JMeter [7],
– maximal throughput of requests – using JMeter.

These tests were done for standard DICOM images (resolution 256x256 and 512x 512). Image file creating was tested for bitmap and JPEG 100% quality formats.

The test took place in a special environment. The Web server was installed on IBM HS20 Blade Server (2x Intel Xeon 2 core 2.8 GHz, RAM 4 GB, FSB 400 MHz) and client was installed on PC computer. These hosts were connected to 1 Gb Ethernet switch. Two operating systems – Linux and Windows Server – were installed on the server. PC client contained an instance of JMeter – testing application. The client PC and server were connected by a 1 Gb Ethernet switch.

5 The Results of Tests

The results of performance test are in Table 1. The data are also shown on Fig. 6, 7 and 8.

Table 1. Time performance of code – operating systems: Linux (L) and Windows (W)

Format of image	Times of: [ms]	Java L	Java W	ASP W
256 x 256	projection	30.5	81.2	42.1
BMP	creation	18.0	46.0	5.0
	response time	94.0	113.0	72.0
	throughput [req/s]	28.3	26.1	30.9
JPG	creation	18.0	46.0	8.0
	response time	85.0	105.0	57.0
	throughput [req/s]	32.5	19.5	42.5
512 x 512	projection	114.7	356.2	171.8
BMP	creation	63.0	124.0	15.0
	response time	370.0	450.0	360.0
	throughput [req/s]	7.5	6.6	4.6
JPG	creation	63.0	139.0	31.0
	response time	331.0	407.0	215.0
	throughput [req/s]	7.8	6.7	9.1

6 Summary

We tested two common web technologies (Java and ASP) running on Linux and Windows in terms of performance for medical image system. The fastest for our system is ASP/C# running on Windows. Although the projection calculating in this case isn't fastest, this technology performed the smallest request time and the highest throughput. The further work is to test such environments as PHP and Mono (.Net platform for Linux) and capability of hardware acceleration for imaging. The goal is to create a web-based transesophageal echocardiography simulator (TEE).

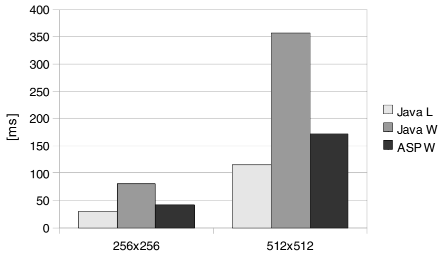

Fig. 6. Time performance of code for calculating a projection

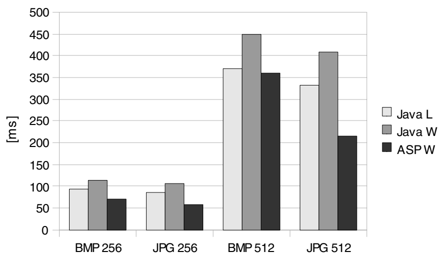

Fig. 7. Response times for single requests with a projection image

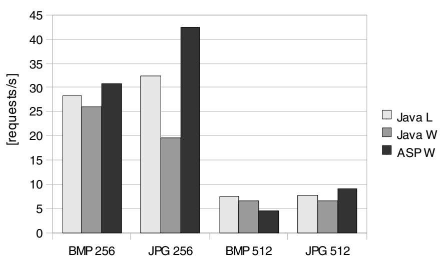

Fig. 8. The throughputs of DICOM image servers

References

1. Sakusabe, T., Kimura, M., Onogi, Y.: On-demand server-side image processing for Web-based DICOM image display. In: SPIE. Proc. 2000, pp. 359–367 (2000)
2. Jianguo, Z., Jianyong, S., Yuanyuan, Y., Chenwen, L., Yihong, Y., Jin, J., Weihua, C., Kun, S., Guozhen, Z.: Image-Based Electronic Patient Records for Secured Collaborative Medical Applications. In: Engineering in Medicine and Biology Society, 2005, pp. 3218–3220 (2006)

3. Ancuta, P.N.: 3D Object Modeling and Visualization Software for Surgery Preoperative Plan. In: 6th Workshop on European Scientific and Industrial Collaboration on promoting Advanced Technologies in Manufacturing, WESIC 2008 (2008)
4. Rosset, A., Spadola, L., Ratib, O.: OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images. Journal of Digital Imaging 17(3), 205–216 (2004)
5. Fernandez-Bayo, J., Barbero, O., Rubies, C., Sentis, M., Donoso, L.: Distributing medical images with internet technologies: a DICOM web server and a DICOM java viewer. Radiographics 20(6), 1817–1818 (2000)
6. Bui, A.A., Morioka, C., Dionisio, J.D., Johnson, D.B., Sinha, U., Ardekani, S., Taira, R.K., Aberle, D.R., El-Saden, S., Kangarloo, H.: OpenSourcePACS: an extensible infrastructure for medical image management. IEEE Transactions on Information Technology in Biomedicine 11, 94–109 (2007)
7. Apache JMeter, `http://jakarta.apache.org/jmeter/`
8. Sample images, `http://rsb.info.nih.gov/ij/download/sample-images.zip`