NetraRIS - a Web based DICOM Viewer

Dinesh B. Kulkarni
Phd, Walchand College of Engineering,
Sangli, India

Mahesh M. Doijade
Walchand College of Engineering,
Sangli, India

Chetan S. Devrukhkar
Walchand College of Engineering,
Sangli, India

Ganesh R. Zilpe
Walchand College of Engineering,
Sangli, India

Rajesh R. Surana
Walchand College of Engineering,
Sangli, India

ABSTRACT

The need of telemedicine is increasing day by day and online support is crucial to have faster diagnosis. In this paper we try to provide an online teleradiology system consisting of 3D medical imaging which enables web based Surface Rendering of DICOM images and efficient radiology information related hospital management and also discusses about challenges in volume rendering on web. Here we carry out rendering at local client using WebGL, so our system takes benefit of its power, due to this processing because of unreliable hindrances such as network failure is eliminated. The system developed exploits the potential of WebGL due to which accelerated graphics pipeline becomes available which otherwise is not. Developed system enables to view surface rendered multiframes DICOM images with no distortion and resolution selection by user to reduce overheads along with features such as zooming, contrast control, color transformation ,rotate, flip, invert and displays DICOM attributes.

General Terms

Medical Imaging.

Keywords

DICOM Standard, Teleradiology, Medical Imaging, Web based Medical Information System, WebGL.

1. INTRODUCTION

A new era of medical imaging had started when Roentgen, the German physicist, developed X-ray photography and since then medical image devices like CT, MRI, etc., began to come into existence. Doctor started using slices of medical images for diagnosis purpose but it was totally based on the experience and excellence of the doctor. Till mid-nineties, no universally accepted standard was available for representation and storage of medical data generated by CT, MRI, etc. Therefore, the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) had developed DICOM [1] (Digital Imaging and COmmunications in Medicine). DICOM provides the integration of different devices like scanners, servers, workstations, printers, and network hardware from multiple manufacturers. In medical imaging, diagnosis techniques like Magnetic Resonance Imaging(MRI), Computer Tomography(CT) produces a sets of slices which are parallel that forms a volumetric dataset these are stored in DICOM standard format. These DICOM objects are managed by PACS [1] (Picture Archiving and Communication System) which provide storage and access for DICOM objects. There are many software packages implementing PACS technology which are available commercially as well as open-source or free. DCM4CHE [2] is one of the open-source software implementing PACS technology implemented using JAVA programming language. It provides archive and toolkit in which archive is used for medical image storage and toolkit is used for query and retrieval. When we retrospect, doctors were using 2D medical images for diagnosis, but as technology advanced, new image processing algorithm were developed along with decreased prices of hardware components. This triggered the new trend of 3D medical imaging which helps to emulate the milieu of real time 3D as of interaction of specialist with the real organs. There are numerous commercial and free standalone software packages e.g. OsiriX [3], Netra DICOM viewer [4]. The Netra DICOM viewer[4] developed previously by our team and hosted on http://sourceforge.net is available for 3D medical image viewing. OsiriX is standalone 3D DICOM viewer which works on MacOS. Netra DICOM viewer is 3D DICOM viewer works on Windows and Linux. However, these standalone software packages compel radiologists to work at one place only. Radiologists are not associated with only one hospital. They always belong to more than one hospital. In case of emergency, Radiologist has to return to respective hospital and complete the job. But if he is able to access the medical data online and build the 3D scene in real time, he can easily save his time and diagnosis will be faster. This can be done by shifting from standalone to online this motivated us to come up with web based solution which overcomes shortcomings of standalone ones. Nevertheless, 3D online medical image viewers are very less as well as they are not mostly open-source or free. Through NetraRIS, we tried to provide a solution for the online 3D medical imaging. NetraRIS is combination of Netra DICOM viewer and RIS (Radiology Information System). This paper is further organized as follows. Section 2 presents Overview of developed web based 3D DICOM viewer along with brief status of the different technologies enabling 3D and volumetric content on web along with our approach. Section 3 presents developed system architecture. Section 4 describes process flow and finally in section 5 we provides some results and section 6 we conclude with stating future work.

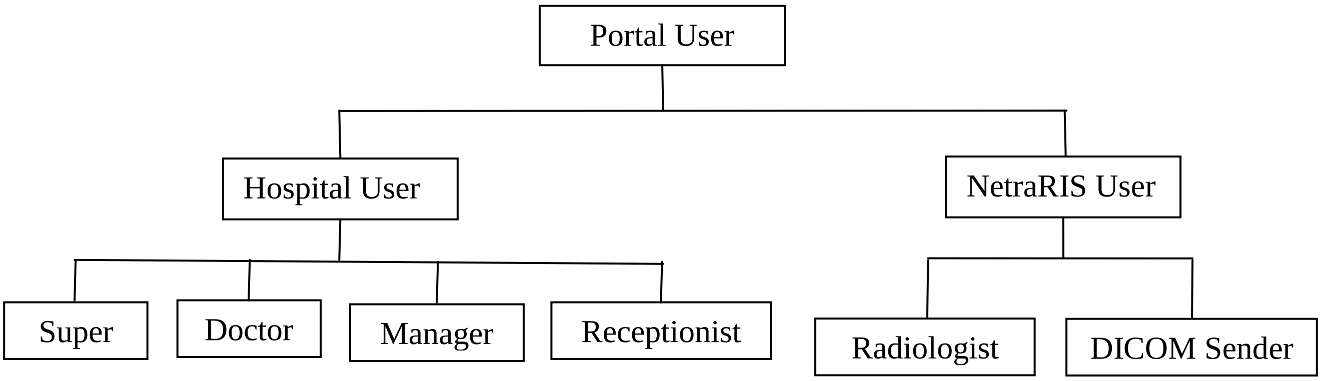

Fig 1: User Hierarchy

2. OVERVIEW OF WEB BASED 3D DICOM VIEWER

2.1 Portal Overview

Using NetraRIS portal, management of radiology related contents of many hospitals can be done easily. The web-portal mainly has two users (see Figure 1), one is hospital user and other is special NetraRIS user. Hospital users include Super, Manager, Doctor, and Receptionist. NetraRIS users are again of two types. First, DICOM Sender is a person who owns or handles CT-SCAN, MRI, etc in particular the lab which generates DICOM (.dcm) images of patient and he uploads DICOM images of related patient to PACS storage. Second is Radiologist, who diagnoses the patient by viewing online 3D representation of DICOM images uploaded by DICOM Sender. Super, a hospital user, grants manager of every hospital and all NetraRIS users. Manager of each hospital grants registration of Doctor and Receptionist. NetraRIS portal provide online appointment facility to patients, on the other hand Receptionist in hospital can also grant appointment to patients who directly come to the hospital. Receptionist sets patient status as 'reported' if patient visit hospital and those reported patient go to the waiting patient list of doctor. Doctor diagnoses the patient and if he needs critical diagnosis then doctor sends patient to DICOM Sender to generate medical images (CT scan, X-ray, etc.) and also assigns Radiologist for that patient.

2.2 Online 3D Generation

Generation of 3D view from DICOM images [1] is a three steps process: to acquire 2D image slices, synthesizing volume data from them, and then rendering that resulting volume data. VTK [5], [6] library can be used to bring this procedure into reality. It is a computer graphics processing and visualization library which can be used for generating 3D views and also exporting/importing them to/from other applications. On the other hand to facilitate online 3D medical imaging, browser should be equipped with WebGL[7] which is open-source, free, and a low level 3D graphics API, uses OpenGL ES 2.0 at its base. Netra DICOM Viewer is open-source standalone software that takes input of the set of DICOM images and renders 3D view. It is fully fledged software that can facilitate Volume Rendering, Surface Rendering, etc. Functions used in Netra [4] strengthen NetraRIS for online 3D representation of DICOM images by using Netra at server side to generate 3D scene. In course of development as we adopt client-server architecture, we tried many different things to transfer and render the 3D scene from Server to Client. Our attempts started with X3D [8], [17] technology. X3D is an open, free and ISO ratified file format to represent and communicate 3D scenes and objects using XML. It uses scene graph technology to describe 3D scenes. But it has visualization problem with web browser. It requires separate plug-in for rendering of 3D scenes and has compatibility issues with some browsers. Next was X3DOM, but it also did not give convincing results. Further we decided to try MedX3D [9], [17], [18] which is an open standard for the visualization of 3D Medical data. MedX3D requires H3DAPI [10] which is an open source haptics development API which uses OpenGL and X3D standards to define graphics. But MedX3D had problem of integration with web browser plug-in provided didn't worked properly. We tried JSON [11] format after MedX3D. JSON (JavaScript Object Notation) is a lightweight data-interchange format, easy to read, write and parse. But there is no way in VTK to export the 3D scene in JSON format. Finally after our all attempts we found that Wavefront obj format [12] is the appropriate candidate. OBJ is an open and free file format developed by Wavefront Technologies which can store and define 3D geometry. Approach for this is to render 3D DICOM scene at server-side, export it in OBJ file format, send it to client and parse it to show 3D scene in client's browser. The OBJ file is created by applying marching cube algorithm [16] onto the set of DICOM images all the requisite data for isosurface generation is stored in this OBJ file. Isosurfaces are effective in representing regions of respective density in 3D CT scan which makes possible visualization of bones, internal organs and other structures we have used AJAX [13] for sending the .obj file to client. For parsing and 3D visualization of this .obj file, we have used Thingiview.js [14] which is JavaScript framework that uses Three.js as its 3D Engine. It supports binary or ASCII OBJ files. It uses HTML5 canvas and WebGL technology for visualization.

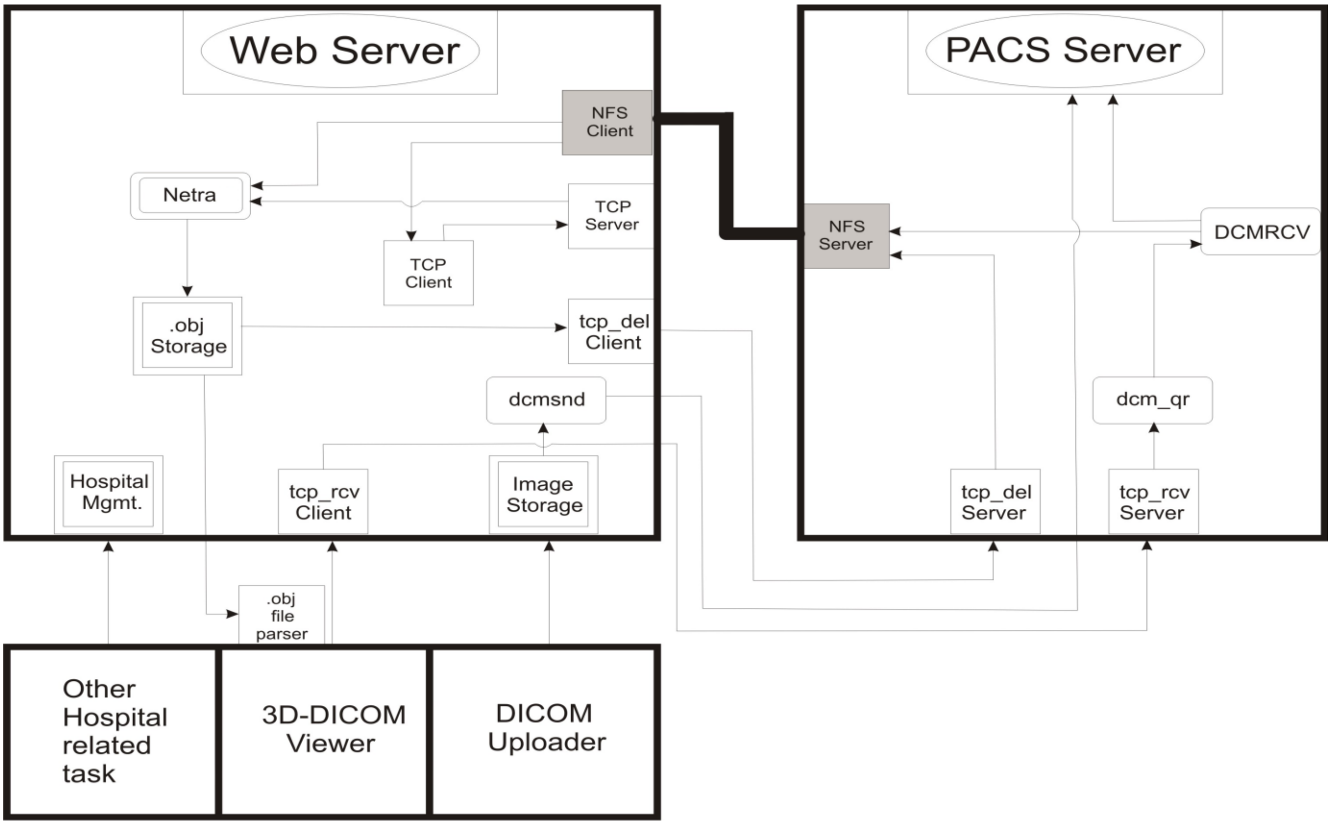

Fig 2: System Architecture

3. SYSTEM ARCHITECTURE

3.1 Web Server

Web server has following modules (see Figure 2):

3.1.1 Netra

Netra is an application written in C++ and using VTK library. It is developed from open source project, Netra DICOM viewer with some modifications. It takes DICOM image folder, creates 3D view and then exports it to wavefront obj format.

3.1.2 tcp_rcv_client

It takes parameters from users like Patient Name, wraps it and sends it to 'tcp_rcv_server' listening on port to receive DICOM images from PACS server.

3.1.3 tcp_del_client

It sends request to 'tcp_del_server' listening on port to delete DICOM images after its use.

3.1.4 tcpclient

It wraps parameters like path of DICOM folder and path to save obj file and sends it to 'tcpserver' listening on port.

3.1.5 tcpserver

It is listening on port. It unwrap the parameters sent by 'tcpclient', set call for Netra and calls it to form .obj file.

3.1.6 NFS-client

The NFS-client remotely accesses the DICOM image folder that is stored on the PACS server (NFS-server). It offers that folder to Netra as input.

3.1.7 dcmsnd

It is part of dcm4che toolkit. It takes the image folder and sends and stores it on the PACS server.

3.2 PACS Server

PACS server has following modules (see Figure 2):

3.2.1 tcp_rcv_server

It is listening on port. It unwrap the parameters sent by 'tcp_rcv_client', set call for 'dcmqr' and calls it to retrieve the DICOM files.

3.2.2 tcp_del_server

It is listening on port. It unwrap the parameters sent by 'tcp_del_client', set call for 'rm' and calls it to delete the temporary DICOM files.

3.2.3 dcmqr

It is part of dcm4che toolkit and used for querying and retrieving DICOM images.

3.2.4 DCMRCV

It is an AE (Application Entity) listening on port. It takes signals from 'dcmqr' and retrieves DICOM images from PACS server and stores it on respective location on disk drive.

3.2.5 NFS-Server

NFS-Server shares the DICOM directory with Web Server through nfs sharing and thus, provides Web Server with DICOM data.

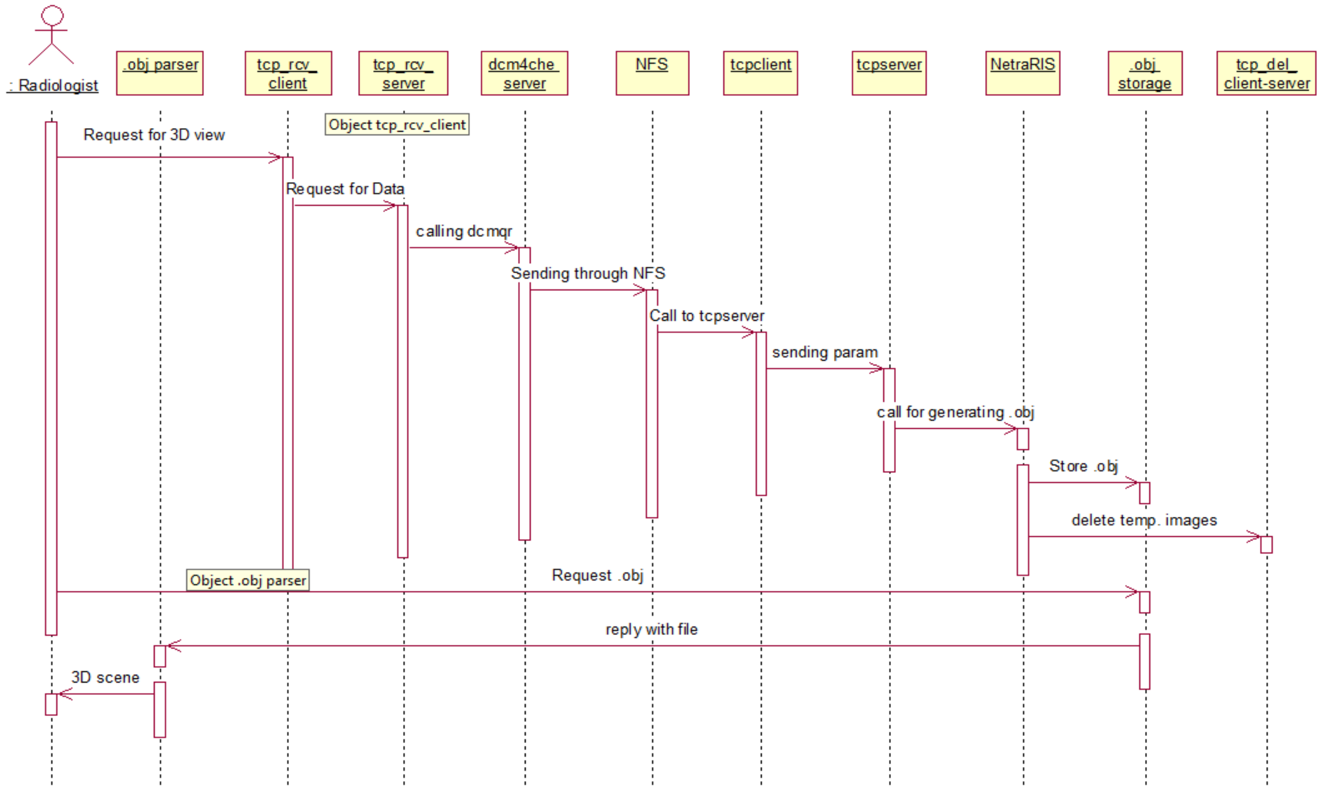

Fig 3: Process Flow

4. PROCESS FLOW

There are two major processes in above NetraRIS system architecture (see Figure 3): uploading and 3D viewing. First is uploading which is used by DICOM Sender for sending and storing medical images of related patient in PACS server. Data is sent to web server using AJAX and stored in a temporary directory. dcmsnd, a part of dcm4che toolkit, is used to send temporary directory data to dcm4chee archive. Second process is online 3D medical imaging. In this, patient’s name is sent to tcp_rcv_client. tcp_rcv_client wraps the parameters and sends to tcp_rcv_server which unwrap parameters and calls dcmqr, a part of dcm4che toolkit. dcmqr queries and retrieves the image data into NFS shared directory which is shared between PACS server and web server. Now web server has medical image data and ready to call Netra .obj exporter. TCP client-server communication is used again to call Netra, because Netra has large size which restricts php system call to invoke it. With the help of TCP and Netra, .obj file is stored in obj storage. NetraRIS 3D DICOM viewer requests .obj file from obj storage using AJAX, parses it in browser and displays 3D output using WebGL which is seen by Radiologist.

5. RESULTS AND DISCUSSION

The system environment for demonstration purpose is as follows, hardware of both the Web Server and PACS Server consists of the following specification: Intel Core i5 @ 3.2 GHz, 4 GB DDR3 RAM, Apache Web Server, MySQL database and Operating System - CentOS 6.1, 64 bit. At Client side hardware specification is as follows: AMD Athlon II X4 630 @ 2.8 Ghz, 4GB DDR3 RAM, GPU Device - NVIDIA GTX 465, browsers used - Firefox, Chrome, Opera, Operating System - Windows 7, 64 bit. Client and Server are connected through LAN. We did not considered Internet Explorer as it still does not supports WebGL though there are non-commercial third party provided plug-in such as IEWebGL [15] . It is recommended to have dedicated PCIE or AGP graphics card instead of onboard graphics for better performance. Figure 4 shows the web based 3D DICOM viewer isosurface version of CT scan of ribs and illustrates some crucial features of the viewer.

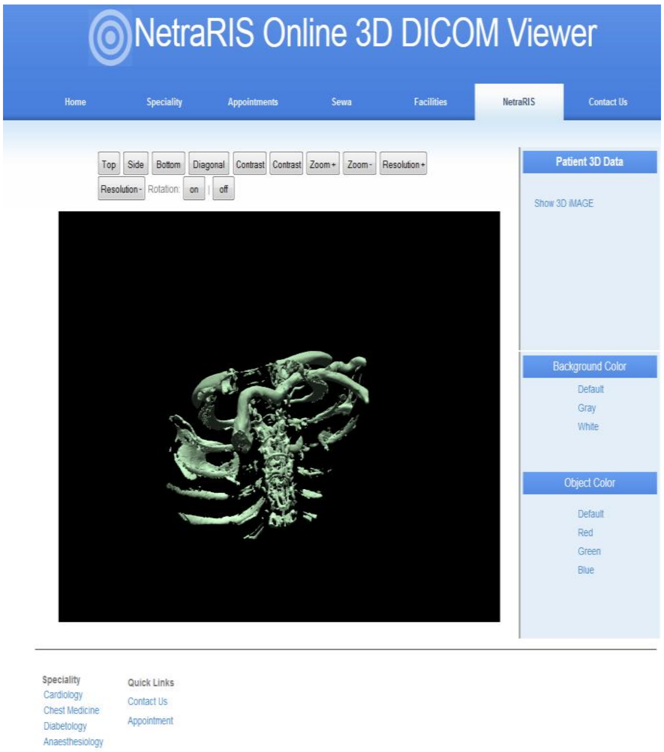

Fig 4: Web 3D DICOM Viewer

The results shown are average of approximately 20 independent runs (see Table 1). Here for demonstration purpose DICOM images for each Cardiac, Ribs, Skull taken are subset of their larger set of DICOM images. It is observed that Chrome's performance is slightly better than Firefox that is mostly due to the level of WebGL support provided by each of them. As medical imaging and its 3D viewing is data intensive as well as compute intensive moderate to higher end CPU, good amount of RAM and GPU Card at client side is highly desirable. We analyzed that generation of OBJ file does not take much time as compared to sending it to the client side, also parsing and rendering 3D scene is very much compute intensive.

Table 1: Results of performance evaluation of the system

Browser	DICOM image	DICOM image/OBJ generated size (in MB)	Generation time of OBJ file at Server + Download time of OBJ at Client (in seconds)	Parsing OBJ & Rendering 3D Scenes time (in seconds)
Firefox 12.0	Cardiac	56.3 / 56.3	31.4	39.67
	Ribs	102.5 / 56.8	35.53	43.77
	Skull	65.3 / 56.6	32.12	41.45
Chrome 19.0	Cardiac	56.3 / 56.3	30.4	37.86
	Ribs	102.5 / 56.8	33.33	40.91
	Skull	65.3 / 56.6	31.44	39.48

6. CONCLUSION AND FUTURE WORK

The system view and results presented here illustrate the potential of complex rendering visualization in web browsers. Such an initiative ignites attraction towards visualization primitives implemented in high-end desktop visualization applications to be ported onto web. The system developed can be effective in representing regions of specific density in three dimensional CT scan which makes possible visualization of bones and other organs. As the system developed carries out parsing and rendering at client side, server's load is distributed and hence considerably reduced. Also this approach empowers with mobility and faster diagnosis to radiologist along with other entities involved thereby accentuating the teleradiology system. Further we aim to carry out complex volume rendering by developing WebGL based library for that purpose. Also some performance optimizations are still needed which we expect to carry out. Another critical component will be integration of surface rendering within volume rendered scenes.

7. REFERENCES

[1] Oleg S. Pianykh - DICOM. A Practical Introduction and Survival Guide

[2] DCM4CHE, http://www.dcm4che.org, (July 2, 2012)

[3] OsiriX Imaging Software, http://www.osirix-viewer.com (July 2, 2012)

[4] Netra DICOM Viewer, http://sourceforge.net/projects/netradicom (July 2, 2012)

[5] William J. Schroeder, Lisa S. Avila and William Hoffman – Visualizing with VTK: A Tutorial, http://www.doc.ic.ac.uk/~dr/teaching/Visualization/VTK_Tutorial.pdf (July 2, 2012)

[6] Visualization Toolkit(VTK), http://www.vtk.org (July 2, 2012)

[7] WebGL - OpenGL ES 2.0 for the Web, http://www.khronos.org/webgl (July 2, 2012)

[8] X3D, http://www.web3d.org/realtime-3d/x3d/what-x3d (July 2, 2012)

[9] MedX3D, http://www.web3d.org/realtime-3d/working-groups/medx3d (July 2, 2012)

[10] H3DAPI, http://www.h3dapi.org (July 2, 2012)

[11] JSON, http://www.json.org (July 2, 2012)

[12] Wavefront OBJ File Format Summary, http://www.fileformat.info/format/wavefrontobj/egff.htm (July 2, 2012)

[13] AJAX: A New Approach to Web Applications, http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications (July 2, 2012)

[14] Thingivew.js - Javascript 3D Model Viewer, https://github.com/tbuser/thingiview.js (July 2, 2012)

[15] WebGL for Internet Explorer, http://iewebgl.com (July 2, 2012)

[16] W. E. Lorensen, H. E. Cline, 1987, "Marching cubes: A high resolution 3D surface construction algorithm", SIGGRAPH 87, Proceedings of the 14th annual conference on Computer graphics and interactive techniques.

[17] John NW, Aratow M, Couch J, et al., 2008, "MedX3D: standards enabled desktop medical 3D", Studies in health technology and informatics 2008 132, 189.

[18] N. F. Polys, N. W. John, D. Evestedt, M. Aratow.,"Unifying ISO Standards for Medical Image Presentation" http://medical.nema.org/Dicom/minutes/WG-11/2011/2011-02-16/Web3D/web3d_Medical_Profile_draft_2010.pdf , (July 2, 2012)