

m3DICOM: A Platform for Mobile DICOM Visualization Based on X3D

Iuliana Ojog
NeonLobster Ltd.
Bloxham Mill Business Centre
Barford Road
Bloxham OX15 4FF, England
oiuliana@neonlobster.co.uk

Miguel Arias-Estrada
INAOE – Computer Science Dept.
A.P. 51 y 216,
Puebla, Pue. 72000, Mexico
ariasmo@inaoe.mx

Abstract—The authors propose a platform suitable for mobile DICOM visualization and segmentation processing, that overcomes some of the current limitations of web-based DICOM viewers by integrating 3D modeling on the cloud and dynamic generating an X3D file for visualization on a mobile device. Web based DICOM viewing in real-time is a desirable application for physicians/radiologists, but the limitation of high volume data and large bandwidth requirements is a barrier for adoption. With the new tablet / smartphone technologies, and the new generation of WebGL-based browsers with GPU accelerated graphic visualization, it is possible to explore possible alternatives. The proposed approach segments the DICOM file on the server side, and delivers a dynamic X3D model to the mobile device for visualization and interaction with the user. Processing is carried out on the cloud, but final 3D rendering is done on the mobile device, with communication bandwidth requirements reduced in the order of 3 to 5 times.

Keywords—DICOM; X3D; WebGL; HTML5; 3D graphics; mobile medical applications; render on the cloud; GPU segmentation on the cloud; X3DOM.

I. INTRODUCTION

There has been extensive research on using the web for medical image processing, remote diagnosis and visualization [1][2]. Unfortunately, the large amount of data of DICOM files limits the applicability on mobile devices. Several applications currently available for mobile device DICOM visualization require to download the whole DICOM file, that could be time consuming for large studies, i.e., 50 Mbytes. On the other hand, the adoption of WebGL [3] and HTML5 standards for the new generation of browsers brings back the interest of web based 3D rendering with WebGL and declarative X3D [5], since it will be supported without plug-ins. Further, WebGL will make possible to render 3D models in real time with the computational capabilities of the new smartphones and tablets. Some attempts to build WebGL-based medical visualization systems have been reported [4], but only considered the X3D model creation from DICOM on the server side for visualization in a remote platform. In this work, we propose an extended approach where the 3D model is a hybrid representation of the DICOM extracted model,

and other features (like 2D slices for close inspection) and a flexible user interface to control position and transparency of 3D features. Our approach gives more flexibility and visual information to the user than previous approaches. The short paper is organized as follows: section II gives an overview of the architecture, from the server and client sides. Section III details the implementation and preliminary results, and we give some conclusions and directions in section IV.

II. PROPOSED ARCHITECTURE

The proposed platform is based on a client – server architecture, which allows doctors/radiologists to visualize, analyze and interact with patient information stored in a DICOM web repository, using a mobile device and a WebGL enabled browser. The general architecture is depicted on Figure 1.

A DICOM server contains a patient database, and applications to process on the cloud the files to extract an X3D model from the data. The 3D model is created by a segmentation and slicing process on the DICOM data. Later, a complex model is assembled and converted to HTML5 format to be visualized on the client side with a WebGL compliant browser. The client device could integrate hardware graphics acceleration for local 3D rendering.

Since the segmentation process could range from simple thresholding to complex segmentation algorithms, the platform is open to integrate GPGPU acceleration.

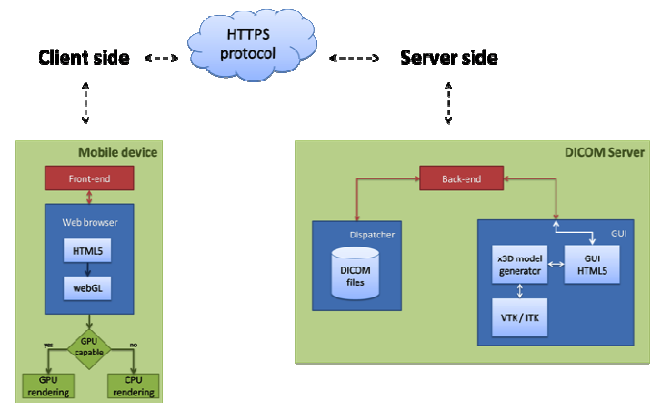


Figure 1. General Server/Client architecture

A. Server processing

Most of the processing is carried out on the server side. The server contains a DICOM repository where the patient data is stored. A file subsystem and access is supported for file upload, storage, management and traditional DICOM access for visualization through the net.

Several medical visualization applications carry out some kind of preprocessing to the DICOM files. The most elaborate and complex processes are based on advanced segmentation algorithms that locate specific organs or tissue in the data. Unfortunately, the processing capabilities of current computers are limited, so cloud processing (i.e., large computer arrays or GPU hardware acceleration on the cloud) has become an alternative to a data center in the hospital. The proposed architecture is open to combine cloud processing for data segmentation and the creation of the X3D model for visualization.

For our proof of concept system, we are using the VTK/ITK libraries for exploring basic segmentation and data slicing on the server side. Furthermore, the dynamic X3D creation is composed of a segmented model of the DICOM data with 2D slices from the DICOM set for close comparison of details by the radiologist/doctor.

The 2D slices are extracted from the DICOM file and combined with the general 3D model as a hybrid visualization model that contains the segmented organ or structures, and 2D slices for close visualization of the original scan. User retains full control on the X3D model control on the web browser, but the slices are dynamically generated and streamed from the server into the web client.

Figure 2 gives more details on the libraries and subprocesses carried out on the server side, and the client WebGL requirements.

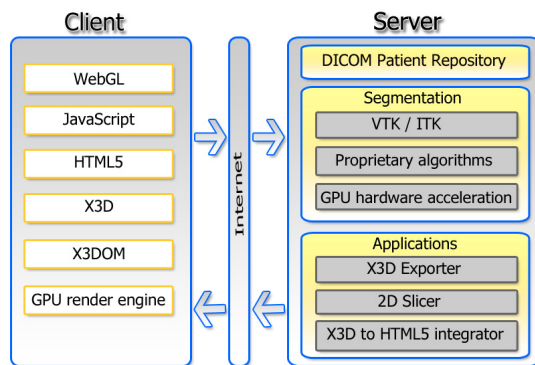


Figure 2. Server-Client architecture of the DICOM/X3D System

B. Client – web user interface

On the client side, we are proposing a web interface using HTML5 and JavaScript. X3D files are integrated into html using embedded X3D as mixed-namespace document (X3DOM implementation [5]), which makes it possible

simple interaction (HTML events) and navigation (zoom, rotate) on 3D objects. The interface is designed to provide cross-browser support (for both desktop and mobiles), using JQuery Mobile framework [6].

Through the user interface, it is possible to set up segmentation parameters (algorithm, thresholding, and reducing factor), which are used by the segmentation application on the cloud.

There are some initial parameters for the X3D object generation, such as transparency, shininess, diffuse color, and so on. These can be later on changed via JavaScript directly on the generated X3D file.

Figure 4 shows a view of the mobile device web interface with a 3D model rendered on the server side.

C. X3D model creation

Figure 3 presents a diagram of the hybrid model creation process. There is a set of default parameters, which the user can update interactively from the web interface. The DICOM file is segmented and a mesh is created. The segmentation process is dependent on the kind of visualization required by the user; for our experiments, we are using simple threshold to isolate bone from other tissues. Once the mesh is created, it is exported as a X3D model.

On the other hand, a set of slices is extracted from the DICOM, and individual X3D models are generated, and integrated into a single X3D model that is then integrated using the X3DOM libraries into HTML5. The selection of the slice is activated based on user interaction by a slide control to visualize the 2D cut details of the 3D model.

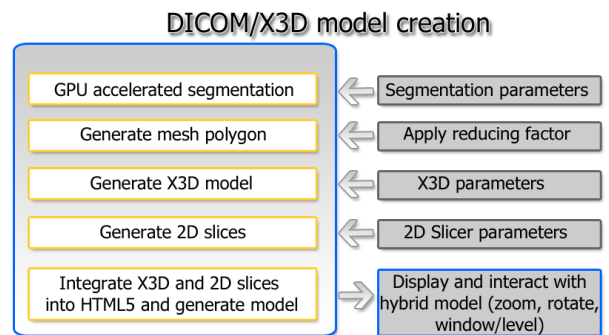


Figure 3. DICOM/X3D hybrid model creation

III. PRELIMINARY RESULTS

An implementation of the platform was carried out. On the server side, the DICOM files are stored in a basic file system, where they can be accessed and processed by a server application. The application communicates with the client to receive the visualization parameters and performs the DICOM segmentation, optimization and delivery of the

X3D model. Processing is carried out using the visualization toolkit libraries VTK [7], ITK [8] and proprietary algorithms used to segment the 3D model from the DICOM data. The application integrates the X3D and 2D slice models into a HTML5 web interface, using JavaScript and declarative X3D.

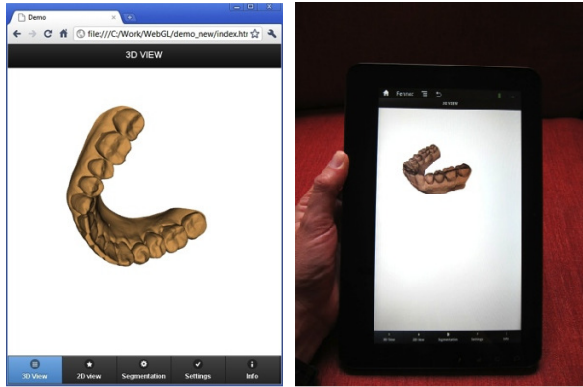


Figure 4. View of the mobile device interface showing a 3D model segmented on the server side.

On the mobile device, the user can access the interface web page through an HTML5 compliant browser. We are using a Viewsonic G-Tablet running Android 2.2 and the Fennec 4.0b6pre browser. Figure 4 shows the interface and the user interaction on the tablet. The Fennec browser still has some limitations (i.e., only single touch event detection) that limits interaction on the image) that had to be hand programmed to allow full interaction with the user.

The following table summarizes the file size and transfer times compared to a pure DICOM download. Since the file size is 2-3 times smaller than a DICOM down-sampled file, the download time is improved in the same proportion, allowing the user to experiment with other segmentation parameters. Furthermore, the user keeps full interaction of the 3D model (transparency, window/level, 3D rotation, zoom) as well as position of the 2D slices that allow the physician to see and verify original details extracted by the 3D segmentation process.

TABLE I. COMPARED SIZES OF DIFFERENT MODELS

Model	Original DICOM size	Reduced size	X3D model	X3D Hybrid model
Dental file	62 MB	10 MB	1 MB	1.5 to 4 MB
Thorax file	20MB	5 MB	500 KB	1.5 to 2 MB
Comments	Too large for mobile	Reduced resolution	Only 3D model	3D model and 2D slices

IV. CONCLUSION AND FUTURE WORKS

We have presented a proof of concept of a hybrid DICOM/X3D visualization platform that efficiently delivers high resolution data to a mobile device optimized bandwidth. The platform visualize a WebGL-based render 3D model from a large DICOM model stored on the server side, with high resolution slices of the scan intersected with the 3D model. The platform is extensible to allow further processing and segmentation on the server side, but flexible enough to allow the mobile user to interact with the visualization and segmentation parameters. In the context of cloud computing, the cloud server will preprocess the DICOM files to generate the X3D models, so they will be ready for user interaction. As future work, we are exploring using GPU based segmentation on the server side, combined with WebGL on the mobile device, to enhance the quality and flexibility of the segmented data.

REFERENCES

- [1] S. K. Yoo, J. Key, K. Choi, and J. Jo, "Web-Based Hybrid Visualization of Medical Images," Image and Video Retrieval, LNCS, Vol 3568. 2005, pp. 588
- [2] A. Poliakov, E. Albright, K. Hinshaw, D. Corina, G. Ojemann, R. Martin, and J. Brinkley, "Server-based Approach to Web Visualization of Integrated Three-dimensional Brain Imaging Data," Journal of the American Medical Informatics Association, 2005 Mar-Apr 12(2), pp. 140-151
- [3] WebGL, „OpenGL ES 2.0 for the Web,“ 2011. Available: <http://www.khronos.org/webgl/> <retrieved: November, 2011>
- [4] D. Cantor-Rivera, R. Bartha, and T. M. Peters, "Efficient 3D rendering for Web-based medical imaging software: a proof of concept," SPIE Medical Imaging 2011: Visualization, Image-Guided Procedures, and Modeling, Volume 7964 Feb 2011, pp. 79643A-79643A-4
- [5] J. Behr , Y. Jung , J. Keil , T. Drevensek , M. Zoellner , P. Eschler and, D. Fellner, "A scalable architecture for the HTML5/X3D integration model X3DOM", Proc. 15th International Conference on 3D Web Technology (Web3D 2010), ACM Press, 2010, pp. 185-193
- [6] JQuery, "jQuery Mobile Framework," 2011. Available: <http://jquerymobile.com/> <retrieved: November, 2011>
- [7] VTK, "Visualization Toolkit, v5. Open Source for 3D computer graphics," 2011. Available: <http://www.vtk.org> <retrieved: November, 2011>
- [8] ITK, "Insight Segmentation and Registration Toolkit, v4. Open Source library of image analysis algorithms," 2011. Available: <http://www.itk.org> <retrieved: November, 2011>