A Survey of Web Based Medical Imaging Applications

Ioan Virag, Lăcrămioara Stoicu-Tivadar
Politehnica University of Timişoara/Automation and Computer Science Faculty
Politehnica University of Timişoara
Timişoara, Romania
ioan.virag@aut.upt.ro, lacramioara.stoicu-tivadar@aut.upt.ro

***Abstract*—Medical imaging applications have an established key role in diagnosis, preoperative surgery planning and medical students training. After the introduction of PACS (Picture Archiving and Communication Systems) and standards like DICOM (Digital Imaging and Communications in Medicine) many software companies around the world provided increasingly better solutions to visualize and share the image sets produced by CT (Computed Tomography), PET (Positron Emission Tomography), MRI (Magnetic Resonance Imaging) or other types of scanners. Often these telemedicine systems require prior installation of specific proprietary operating systems or software components. In response to such shortcomings several open source applications where developed, with an increasing focus on the web based solutions, that eliminate the need to install additional software components on the end users computer or mobile device. This kind of applications not only reduce the installations costs, they also come with the advantage of mobility since they can be accessed by any modern web browser on a suitable smart phone or tablet. Therefore we consider that the solutions presented in this paper will be a great practical value to the involved medical staff.**

Keywords—medical imaging; viewer software; DICOM; PACS; Telemedicine;

I. INTRODUCTION (*Heading 1*)

Medical imaging applications allow physicians to visualize and share patient related data obtained from a prior CT (Computed Tomography), MRI (Magnetic Resonance Imaging), PET (Positron Emission Tomography) or other type of scanning techniques. The resulting files are stored in PACS (Picture Archiving and Communication Systems) using the universally adopted DICOM (Digital Imaging and Communications in Medicine) standard [1].

The systems presented in this paper allow doctors to interact even remotely with their colleagues or medical students during the patient diagnosis, surgery planning or for clinical analysis, facilitating the understanding and study of different scenarios. The web oriented architecture also provides the means to access the applications from any mobile device that has a suitable browser installed. The importance of researching new open source imaging techniques has already been emphasized in [2, 3]. The present paper is taking a step further into describing the new system architectures that are possible due to recently developed web technologies and libraries such as HTML5 (fifth revision of the HyperText Markup Language), WebGL (Web Graphics Library), WebRTC (Web Real-Time Communication), NanoDICOM, JSARToolKit (JavaScript Augmented Reality Toolkit) and ThreeJS.

II. WEB TECHNOLOGIES AND LIBRARIES

The web oriented approach would not be possible without the some of the major web technologies and libraries used in the creation of any modern web page. These technologies allow end users to remotely access other computers from smart phones or tablets in real-time without installing any additional proprietary software components, providing a valuable tool in cases where the needed specialists can be located only in other remote areas.

A. HTML5

As of October 2014, HTML5 is the fifth major revision of the HyperText Markup Language, developed by the W3C (World Wide Web Consortium) and the WHATWG (Web Hypertext Application Technology Working Group). The specification represents the main technology markup language of the World Wide Web and introduces new application programming interfaces that are better suitable for the development of more complex web applications. With the aid of the newly introduced syntactic elements such as <video>, <audio> and <canvas>, HTML5 can handle multimedia and graphical content without relying on other proprietary interfaces, paving the way to use such features even on mobile devices that have a suitable browser. Since HTML5 was drafted as a modular specification, some of the technologies initially included inside it, become part of separate standards such as Web Storage, WebSocket and WebRTC.

B. WebGL

WebGL is a cross-platform JavaScript API, based on OpenGL ES 2.0 (OpenGL for Embedded Systems), that allows the rendering of 2D and 3D computer graphics inside a compatible web browser without the use of any third-party plug-ins. It has been developed and is maintained by the Khronos Group with the participation of the major browser vendors like Google, Mozzila, Apple and Opera who are part of the WebGL Working Group.

WebGL uses the HTML5 canvas element as a drawable region inside a web page and is accessed using Document Object Model interfaces. This way the control code is usually written in JavaScript and the shader is expressed in GLSL (OpenGL Shading Language) that is executed on a computer's GPU (Graphics Processing Unit) giving software developers a more direct control of the graphics pipeline.

C. WebRTC

WebRTC is an open source project that facilitates browser-based real-time communication without the use of any other plug-ins. The framework is being standardized on the API level by the W3C and on the protocol level by the IETF (Internet Engineering Task Force).

At the time of writing this article WebRTC is implemented in the latest Chrome, Firefox, Opera desktop and mobile browsers and it consists of three major components that are accessible using JavaScript:

- *MediaStream* (generated by *getUserMedia*), allows the browser to get access to the user's camera and microphone in order to capture associated media streams.
- *RTCPeerConnection*, is an interface that handles the direct streaming of video or audio data between two peers.
- *RTCDataChannel*, enables the exchange of arbitrary data between two peers of a connection.

WebRTC has been already implemented in web applications like vLine, that is a browser-based video chat application and Sharefest, that allows two peers to share their files without using a third party service to accomplish such a task.

D. NanoDICOM

NanoDICOM is a lightweight, open source DICOM file parser written in PHP. The library allows the extraction of the attached medical data and images that can be visualized afterwards inside a web browser. This could be a key component for any online DICOM viewer as it has been already implemented in various applications.

E. JSARToolKit

JSARToolKit is an augmented reality library written in JavaScript, which allows video tracking of a physical marker in real time across frames. This allows the user to draw a 3D model on top of the detected marker with the help of HTML5 <canvas> element and to move the 3D object as the marker moves. The library was derived from C ARToolKit, a tracking library written in C, which can calculate the distance from the camera to several markers in the same time.

F. ThreeJS

ThreeJS is a cross-browser JavaScript API that allows the rendering and the animation of 3D models inside a web browser without using third party plug-ins. The library reduces the code needed to render objects using WebGL (with the help of the WebGLRenderer object) and it has fallback renderers for older browsers as well (using CanvasRenderer o SVGRenderer). The developers can also take advantage of the many other features of the library such as: built in camera types, textures, materials, lights and shaders used for post-processing.

III. Related Web based Sollutions

One of the first systems that used the Internet in order to transmit DICOM images to a remote mobile device was proposed in 2010 as described in [4]. The lack of many of the web technologies described above allowed only the access to the DICOM files via a remote desktop application and for this reason the telemedicine system even if it was a promising solution could not benefit of the many advantages that a web oriented architecture can provide. The system has been successfully used in the treatment of neurosurgical emergencies in cases where stroke specialist where not available on spot and could be contacted only remotely. The system also provided a basic protection of the personal informations of the patients by blocking the direct transfer of the files and allowing only the visualization of the images from the remote device.

In 2012 an online dental image management and viewer system [5] was proposed that already harvested the power of HTML5 features. The PACS consist of a user management module on the server side which allows adding users, a session based authentication and the conversion of DICOM files into JPEG images that could be displayed on the client side. The resulting API allows the application to run either as a web application or a native mobile application. The web application provides features such as DICOM tag reader, zooming, panning, window/level adjustment and measurement tools based on the scale of the retrieved image. The presented iPad application has the same overall features with the advantage that it could be accessed offline after the images where downloaded and stored on the device.

Another OCS (Online Consulting System) [6] was proposed in 2013 with the focus on coronary angiograms. The system allows the upload of DICOM images to a server by a secured 128 bit VPN (Virtual Private Network) that transforms the images before rendering them inside the web browser using HTML5. The main challenge the team faced was to transform the images to a FLV format without compression in order to obtain the high quality movies needed during the investigation process.

A DICOM viewer [7] based on Dicoogle was proposed in 2013. The system uses a new approach based on HTML5 called ZFV (Zero-Footprint Viewer), underlining the fact that the end user can access the viewer from a pure web oriented application without additional browser plug-ins. The new approach introduced by the Dicoogle indexing engine was that the queries can be executed over a set of distributed DICOM storage locations, which where logically indexed as a single federated unit. This way the users could access several Dicoogle nodes simultaneously, based on their search criteria with a single query. The developers, as they mention it in their article, implemented the following tools in their application:

- "distance, surface, and angle measurement;
- graphical annotations;
- cine-playback of multi-frame sequences;
- export of DICOM files to local disk;
- window/level manipulation;
- synchronized frame navigation;
- image rotation and flipping;
- image filtering;
- multiple presentation layouts;
- full screen and dual screen mode;
- zooming and panning."

Our team proposed another browser-based medical visualization system [8] in 2014 suitable for patient diagnosis and training of medical students. The application could reconstruct and render a 3D model based on the patients DICOM files, superimpose that model on the investigated body area using JSARToolKit and share it remotely using WebRTC. This new approach enables physicians to see both the patient's physical image and the underlying tissue structure, which could be a valuable tool in comparing the subject's current status against a previous investigation or in explaining specific problems to the students. The main advantage is that virtually attaching the 3D model to the specific area of the body allows the model to rotate and move in the same time with the patient. As we can see in Fig.1 the system requires only a few steps to achieve the final reconstructed model and almost every step is automated already.

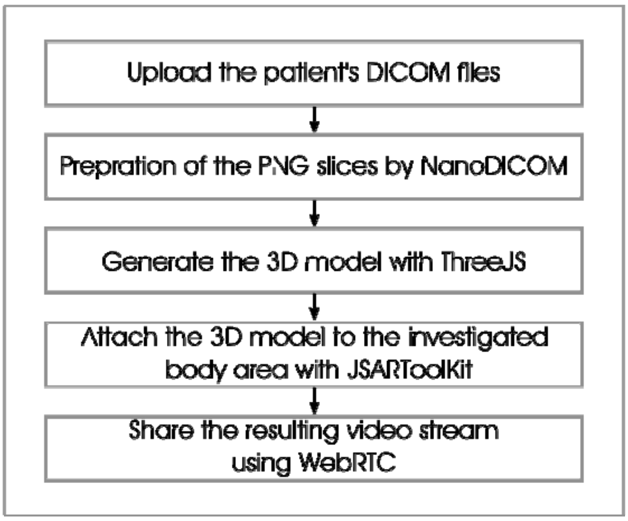

Fig. 1. The steps required to obtain the final 3D model.

We used the NanoDICOM parser to extract the imagistic data from the DICOM files and to provide the individual slices as transparent PNG images that can be displayed in the browser. We also developed a segmentation algorithm in order to isolate only the relevant parts of the slices as in Fig. 2.

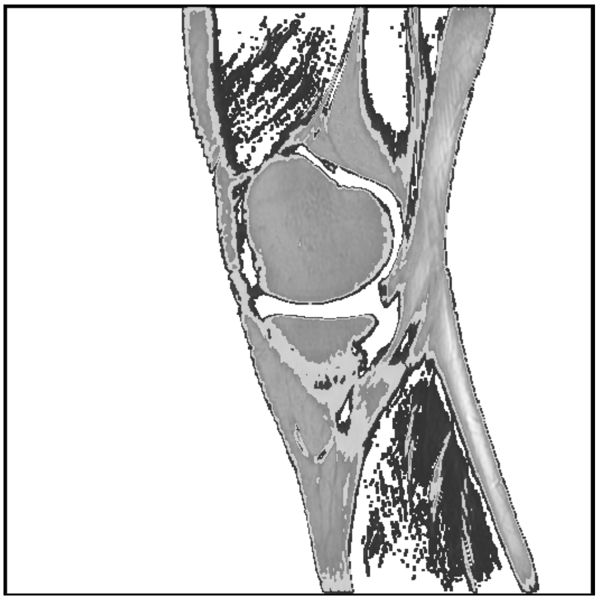

Fig. 2. The resulting processed image slice.

This automatic segmentation phase was necessary in order to remove the surrounding background artifacts produced by the scanner during the image acquisition.

For the reconstruction phase we used the ThreeJS library that allowed us to apply each PNG image as a 2D texture to a PlaneGeometry object and reconstruct a 3D model of the patient's area of medical interest. The application also allows the user to highlight each individual slice of the reconstructed 3D object by modifying its opacity in cases where specific areas require special attention.

The resulting combined image can be afterwards streamed to a remote browser with WebRTC where another physician could further investigate the patient's condition and provide his valuable second opinion.

The proposed system is still under development and further investigation is needed to implement distance or angle measurement or an online image editor for the colorization of the slices. For now we already started working on the colorization module that will allow physicians to colorize the slices based on the tissue density or to isolate specific parts by selecting only a region inside each slice. Our team also intends to further improve the segmentation algorithm that could provide better slices for the reconstruction module.

IV. Conclusion

The cross-platform, open source libraries and APIs can have a significant role in the development of web-based medical imaging solutions. In this paper we described several medical visualization systems that can compete with their proprietary equivalents and allow physicians and medical students to access the applications in a mobile environment. The underlying web technologies also can significantly improve the current viewing platforms using the new web oriented APIs and libraries.

ACKNOWLEDGMENT

This paper is supported by the Human Resources Development Programme POSDRU/159/1.5/S/137516 financed by the European Social Fund and by the Romanian Government.

REFERENCES

[1] DICOM homepage, available at: http://medical.nema.org. Last accessed on April 2nd, 2014.

[2] P. Nagy, "Open Source in Imaging Informatics," Journal of Digital Imaging, vol. 20, Suppl. 1, pp. 1-10, November 2007, DOI: 10.1007/s10278-007-9056-1.

[3] T. Karopka, H. Schmuhl and H. Demski, "Free/Libre Open Source Software in Health Care: A Review," Journal: Healthcare Informatics Research, 2014, Volume 20, Number 1, Page 11, DOI: 10.4258/hir.2014.20.1.11.

[4] Y. Shibata, "A remote desktop-based telemedicine system," Journal of Clinical Neuroscience, Volume 18, Issue 5, May 2011, pp. 661–663, DOI: 10.1016/j.jocn.2010.10.005.

[5] K. R. Saikaew et al., "CT Image Management and Communication Services," 2012 Service Research and Innovation Institute Global Conference (SRII), July 2012, pp. 660 - 666, DOI: 10.1109/SRII.2012.79.

[6] C. Jenei, T. Ivanfai, B. Koszegi, Z. Koszegi, "Webbase Online Consulting System for quasi real time consultations of cardiac images," Computing in Cardiology Conference (CinC), September 2013, pp. 975 – 978.

[7] E. J. Melicio Monteiro, C. Costa, J. L. Oliveira, "A DICOM viewer based on web technology," IEEE 15th International Conference on e-Health Networking, Applications & Services (Healthcom), October 2013, pp. 167 - 171, DOI: 10.1109/HealthCom.2013.6720660.

[8] I. Virag, L. Stoicu-Tivadar, E. Amăricăi, "Browser-based medical visualization system," IEEE 9th International Symposium on Applied Computational Intelligence and Informatics (SACI), May 2014, pp. 355 - 359, DOI: 10.1109/SACI.2014.6840092.