

# 3D Markup of Radiological Images in ePAD, a Web-Based Image Annotation Tool

Dilvan A. Moreira\*, Cleber Hage<sup>†</sup>, Edson F. Luque  
 Department of Computer Science  
 University of São Paulo  
 São Carlos, Brazil  
 Email: \*dilvan@usp.br <sup>†</sup>cleberhage@gmail.com

Debra Willrett<sup>‡</sup>, Daniel L. Rubin<sup>§</sup>  
 Department of Radiology  
 Stanford University  
 Stanford, USA  
 Email: <sup>‡</sup>willrett@stanford.edu <sup>§</sup>dlrubin@stanford.edu

**Abstract**—Quantitative and semantic information about medical images are vital parts of a radiological report. However, current image viewing systems do not record it in a format that permits machine interpretation. The ePAD tool can generate machine-computable image annotations on 2D images as part of a radiologist's routine workflow. The tool has been evaluated in image studies with good results. Since ePAD currently only provides 2D visualization and annotation of images, we developed a plugin to ePAD for the visualization of volumetric image datasets, using the three planes: axial, frontal and sagittal. A study with 6 radiologists was carried out to determine the best interface for also marking 3D ROIs. Video prototypes were created for 3 options: join pixels based on intensity similarity; detect borders around image features; and paint ROIs using a spheric 3D cursor. The 3D cursor was the preferred option. We present these results and also show the final 3D cursor implementation.

**Index Terms**—3D image annotation, radiological reports, tracking lesions, 3D web interfaces, WebGL, cancer, ePAD

## I. INTRODUCTION

Radiographic images play an important role in the diagnosis, prognosis and treatment of cancer. They are used by health care providers and during research trials of new drugs. Images are used to identify location, extent and features of tumor lesions in qualitative (e.g. lesion density) or quantitative (e.g. lesion dimensions) ways. The interpretations of these images will inform the disease state, again in qualitative, "partial response to treatment", or quantitative, "50% decrease in tumor burden" ways. Oncologists use images to determine clinical stage and guide initial treatment decisions. They also use them to monitor response to treatment; duration and changes in therapy; and recurrence or progression of lesions [9].

A key problem in this whole process is that radiology reports are recorded as non-structured free texts, which makes it impossible to perform even simple calculations of changes in lesion dimensions over time or estimates of tumor burden (those estimates are based upon lesion measurements). Free text reports also lack a reference to the image region of the findings to which they refer, and they are not machine-computable. This is a challenge in clinical trials, where data about patient lesions, such as tumor burden, have to be consistent to be integrated and statistically analyzed.

One solution is to use image systems that support Annotation and Image Markup (AIM), a standard developed

by the National Cancer Institute (NCI) [3]. AIM provides an information model for storing the key information about lesions needed to describe lesions and to apply quantitative criteria of response assessment, such as lesion identification, location, size measurements, method of measurement, and other quantitative features.

The electronic Physician Annotation Device (ePAD)<sup>1</sup> is a web-based freely available software platform that uses AIM to store image measurements to enable interoperability [15]. A primary design consideration for ePAD was the storage of image annotations and markups. Annotations describe the results of image interpretation. These annotations can include the imaging time point, the name, type, anatomic location, and measurement of target lesions, the coordinates of the line drawn to make the measurements, and the identifier of the image measured. Image markups are the visual representation of the annotations (e.g., a line drawn on the image indicating a lesion measurement).

ePAD facilitates the radiologist in carrying out lesion measurements as part of a routine image interpretation workflow. Also, because it records all image measurements and annotations in AIM, it is possible to repurpose image data for analysis using alternative imaging biomarkers of treatment response. ePAD uses an easy-to-use web interface that mimics the looks of more conventional desktop imaging applications. This approach joins the advantages of a web application, in terms of administration and installation, to the familiarity and responsiveness of desktop applications.

The ePAD platform (and a desktop-based predecessor called iPad [13]) has been successfully deployed in the workflow to support many research studies [5], [6], [12]. In [14], a radiologist evaluated computed tomography (CT) imaging studies from 20 subjects having one baseline and three consecutive follow-up imaging studies with and without ePAD. This was done to estimate the impact of ePAD on radiologist efficiency in quantitative assessment of imaging studies. The radiologist made measurements of target lesions in each imaging study (using the RECIST 1.1 criteria [4]) with and without ePAD (after a 30-day washout period). The mean total time required to review the images and summarize measurements of target

<sup>1</sup><http://epad.stanford.edu>

lesions was 15% ( $P < .039$ ) shorter using ePAD. In addition, it was possible to rapidly reanalyze the images to explore lesion cross-sectional area as an alternative imaging biomarker to linear measure.

At this stage, the major disadvantage of ePAD is that it currently only provides 2D visualization and annotation of images. To solve this problem, it was decided to move ePAD to a plugin-based architecture. In this paper, we present the ePAD plugin to visualize volumetric image datasets. The code for this plugin was developed at the Department of Computer Science (University of São Paulo) with evaluation input from the Department of Radiology (Stanford University). We also present a study with 6 radiologists that shows that a spherical 3D cursor is the preferred interface to create 3D ROIs.

## II. RELATED WORK

An increasing number of tools are supporting AIM, including open source projects such as iPad plugin for OsiriX [13], AIM ClearCanvas [7], and 3D Slicer AIM [1]. There are also some commercial tools using AIM that are in development [17]. Of the open source projects, none is available as a Web application and only the 3D Slicer tool can show images in a 3D way. Apart from the iPad plugin (a predecessor of ePAD), we could not find in the literature any information about actual use of the AIM ClearCanvas or 3D Slicer AIM in a medical setting.

The ePAD platform comprises two main parts: (i) the client ePAD viewer, a Web-based image viewer and AIM annotation editor that provides a graphical user interface for viewing and recording image interpretations, measurements, and other metadata (Figure 1), and (ii) the ePAD Web Services application, which provides a programming interface to the ePAD server resources: an image database, an AIM annotation database, and analytical plugin modules for processing images.

To add 3D capabilities to ePAD and easy future development, it was decided that the ePAD viewer would be rewritten as a framework capable of hosting plugins. These plugins would implement whatever GUI capability needed for the platform. Following this strategy, the ePAD viewer 2D code was transformed in the 2D plugin and this 3D plugin created.

## III. EPAD 3D PLUGIN

The ePAD 3D plugin shares the same design goals of ePAD, to have a familiar desktop look-and-feel for radiologists and to be easy to use. Following that, we decided to use the traditional 3D Orthogonal Multi-planar Reconstruction (MPR) interface (found in many desktop image tools, such as OsiriX or ClearCanvas). It allows the display of a 3D representation of the volume using 3D orthogonal MPR slices or planes: the axial, frontal and sagittal planes. Using this interface, users would be able to draw 2D (i.e. lines, polygons, circles) and 3D Regions of Interest (ROIs). The problem was divided in 3 parts:

- 1) Create the 3D MPR interface to show the images;
- 2) Decide which kind of interface to use to draw 3D ROIs;
- 3) Implement the drawing interface.

### A. 3D MPR Display Interface

The main problem creating the 3D MPR web-based interface is performance. There are two bottlenecks to get desktop-like performance in a web browser for this interface: the speed that images can be downloaded from the servers and the speed the browser can process them. A typical image series can have between 100 to 400 images, each image has 512x512 pixels, and each pixel has 16 bits (for CT and PET scans). Downloading uncompressed DICOM [11] images would take too long, and DICOM is not a format supported natively by browsers. Users will tolerate some delay loading images, but not much. On the processing speed front, web browsers use Javascript to process data. Javascript is a slow scripting language, not well suited to work with image processing.

To reduce the size of image data being sent over the network, ePAD converts the DICOM images to PNG, a lossless image format that the browser decode natively (and fast). To reduce the number of server requests, images are packed in bundles of 16: 4x4 images bundled together to form a bigger 2048x2048 pixels image. Even with this solution, to send images for the 3 planes would take too long. So, we only send the axial plane original images and generate the other two planes, on the client side, by interpolation. Unfortunately, doing this using Javascript would be too slow. The solution was to use the new HTML5 JavaScript library WebGL [10].

WebGL allows calls to the OpenGL engine [16] running in the graphic board (or chip) in the machine the browser is running, in a portable way. It also allows running programs directly in these boards. OpenGL use is widespread, it can be found in smartphones, tablets or desktop computers. Using WebGL, the programs needed to generate the frontal and sagittal planes can be run on the graphic boards at very high speeds.

After solving these problems, the implementation was straightforward, leading to the interface shown in Figure 2.

### B. Interface to Draw 3D ROIs

In the case of interfaces to draw 3D ROIs, there are no widely used interfaces. To decide which interface to adopt, a study was carried out to determine the best interface for marking 3D ROIs. After discussions with collaborators from the radiology field, three possible interfaces were chosen:

- 1) Click-a-point in an image and join pixels around it based on intensity similarity;
- 2) Detect borders around image features;
- 3) Paint ROIs using a spherical 3D cursor.

To show collaborators how these interfaces would look, we created video prototypes of the three interface options in the new 3D interface of ePAD. A video prototype shows user interactions with an interface that does not yet exist. In our case, we used ePAD's new 3D interface as background and added drawings from image editing tools, such as Gimp.

In the first interface, pixel color similarity is used. In the video showing it, after a image series is opened, a user clicks in a point in the axial plane, then all pixels with intensity

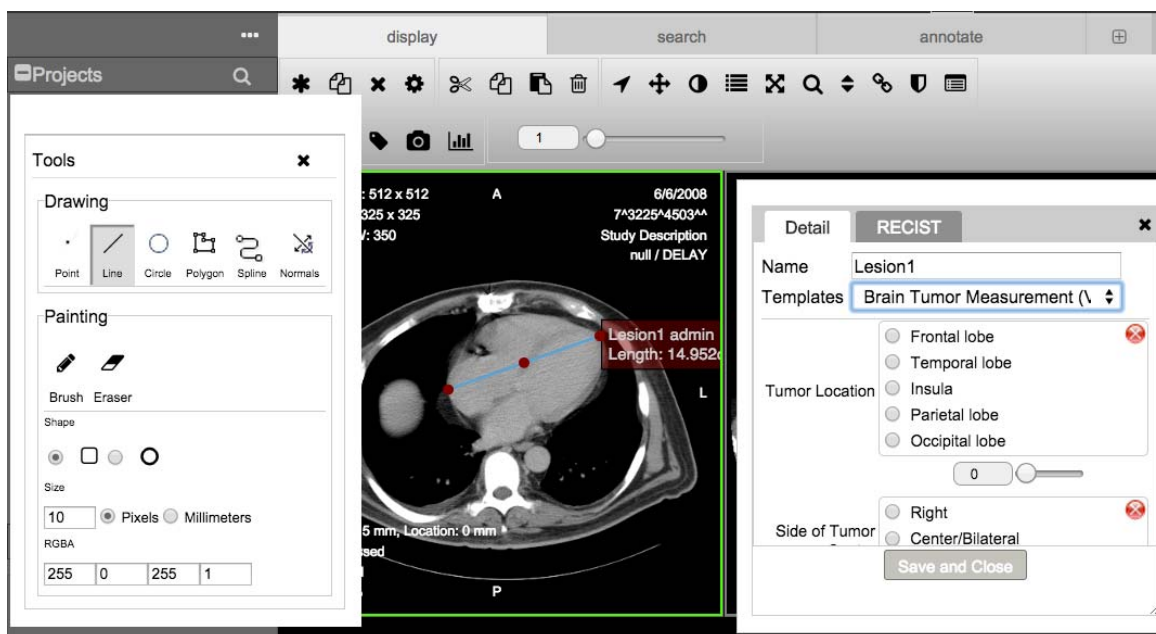


Figure 1. **Client ePAD 2D viewer.** It can only show one plane (center). In this example, a line has been drawn in the image (line tool selected in the left popup) and terms are being associated to it (using the right popup). If the radiologist forgets to talk about some required aspect of the lesion, that aspect will remain in red and he will not be able to save his annotations.

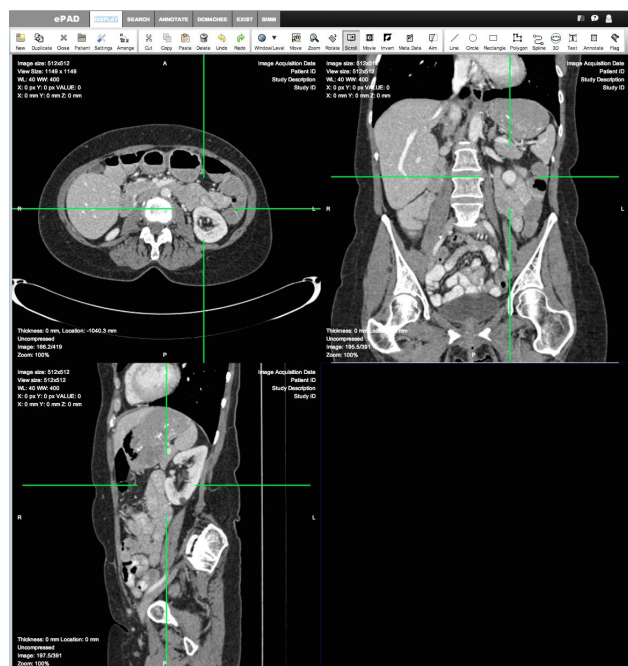


Figure 2. **3D Orthogonal MPR display.** The 3 planes are shown: axial (upper left), frontal (upper right), e sagittal (bottom left). The green hair crosses show the position of the other two planes and can be moved using the mouse, causing the positions (and images) of the other planes to change.

values similar to the one chosen by the user are also selected. A similar pixel has an intensity value that differs from the chosen pixel by less than an interval controlled by the user. By changing this interval, a user can increase or decrease the selected area. Not all pixels in an image will be selected in this way; there is a radius (also user controlled) where this operation takes place. This method of pixel selection is used by drawing tools, like Gimp, to select image areas<sup>2</sup>. Figure 3 shows an area being selected using this method. In the video, it is explained that this method can be applied to just one slice or also to adjacent slices (allowing a 3D selection).

The second interface, border detection, is similar to the first one. A user selects a point and a border detection algorithm is used to find the limits of a region around this point. Figure 4 shows the detail of an image where a user selected a point inside a darker spot that is now almost surrounded by the no-circular green line. The border detection algorithm was able to draw a border around most of the darker spot. A weakness of this method is that border algorithms do not always find closed borders around regions. To couple with that, a cursor (yellow cross in the figure) can be used to draw lines to close open regions by hand. The green circle around this cursor shows the maximum distance, from a selected point, that the border algorithm should look for a border. This distance can be changed by the user. This method can be used in just one image in a series or in adjacent images at once.

In the third interface, a 3D cursor, in the form of a sphere, is used to paint pixels in multiple adjacent images at the same

<sup>2</sup><http://docs.gimp.org/en/gimp-tool-fuzzy-select.html>

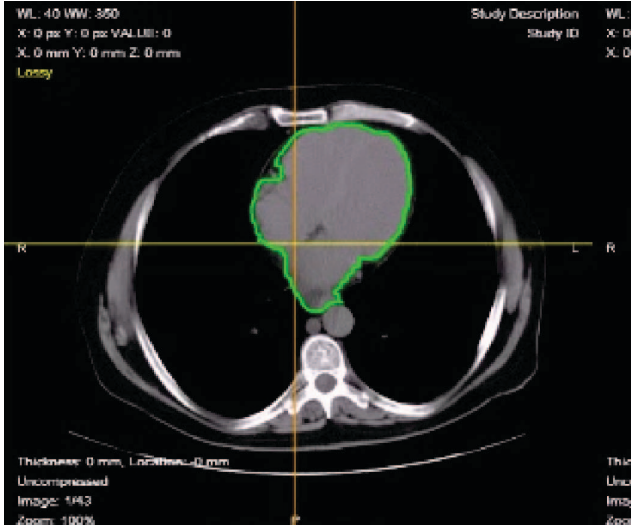


Figure 3. **Intensity Similarity interface:** A user clicked in a pixel inside the marked area (in green), then he changed the intensity interval where other pixels are considered similar to the first (and thus part of the marked area). He continued to do that until he was happy with the selected area.

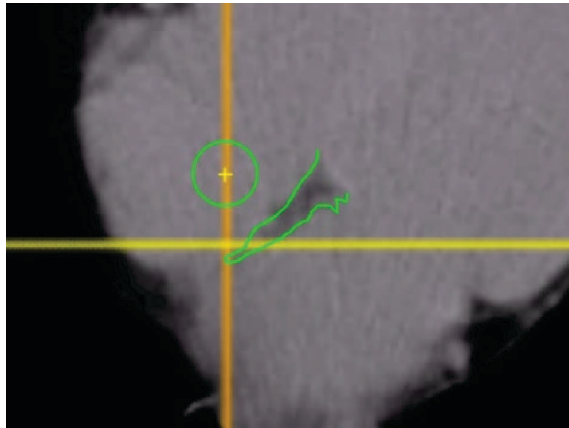


Figure 4. **Border Detection interface:** A user clicked in a point inside the dark spot now almost surrounded by the open green line (center). The border detection algorithm was able to detect most of the border around the dark spot. A cursor (yellow cross) allows the user to close the borderline by hand.

time. Figure 5 shows a red sphere representing the 3D cursor. It projects a set of circular ROIs, with different radii, in each layer on an image series. Users can change the position of the cursor in the 3D space and its size (sphere radius) thus changing the sphere projections in each layer it touches. Figure 6 shows an ROI marked by the cursor (in red), the ROI is a sphere but it appears with different sizes in different planes as the planes have different zoom settings. Using this cursor as a 3D brush, users can paint or clear pixels in different image layers at the same time creating 3D ROIs. This kind of 3D cursor is used in some commercial image applications<sup>3</sup>.

<sup>3</sup>Such as MIM: <http://www.mimsoftware.com>

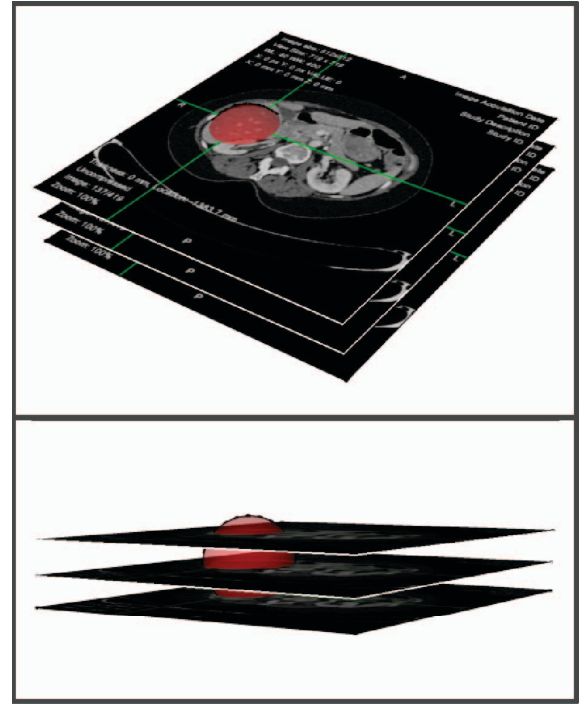


Figure 5. **3D spheric cursor concept:** an imaginary red sphere represents the 3D cursor. It projects a set of circular ROIs, with different radii, in each layer on an image series.

### C. Study with Users

After creating the three video prototypes, a study was carried out with 6 radiologists, with different backgrounds, to evaluate them. Two radiologists were professors from the Department of Radiology at the Stanford University School of Medicine, two were radiology medical students from the same department, and two were professors of radiology at the University of São Paulo School of Medicine at Ribeirão Preto. No user had experience with other tools to mark 3D ROIs in medical images. All users did not have problems seeing the videos or reading the instructions. Their inputs were taken using a questionnaire and interviews (either using video conference or in person).

The prevailing view was that the 3D spheric cursor was the best choice. That is the more manual of the three interfaces, but the experts expressed doubts about how useful would be the semi-automatic approach of the other interfaces. They feared that the algorithms, used to determine pixel similarity and detect borders, would not be precise enough to detect lesion regions most of the time, leaving a lot of manual work to be done by the radiologists. Some of the radiologists may have known the 3D spheric cursor interface from other tools, but none have actually used it before. On the other hand, some of them suggested that we should try the other two interfaces (or some variation of them) as a second way of marking 3D ROIs in a future version of ePAD.



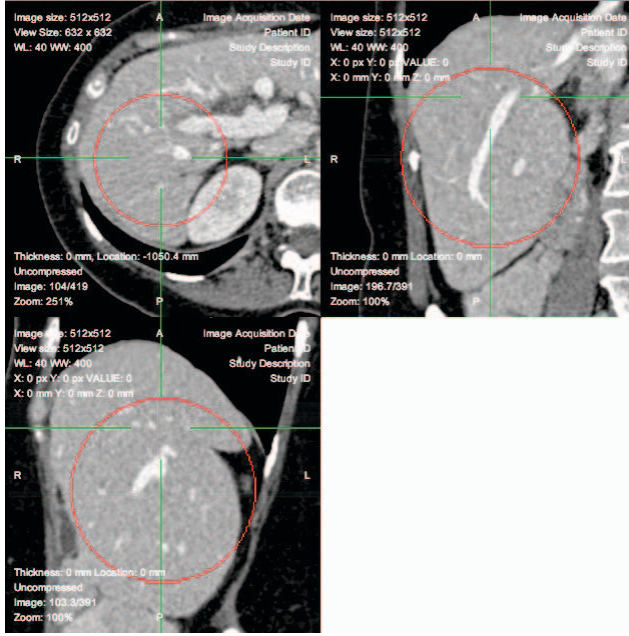


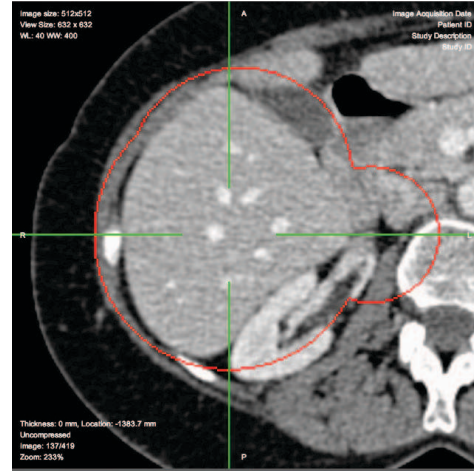
Figure 6. **Spheric ROI**: the interface shows its projections on the different planes (red circles). The projections radii will vary depending on the current plane zoom and where the plane intersected the ROI sphere.

#### D. Spheric Cursor Implementation

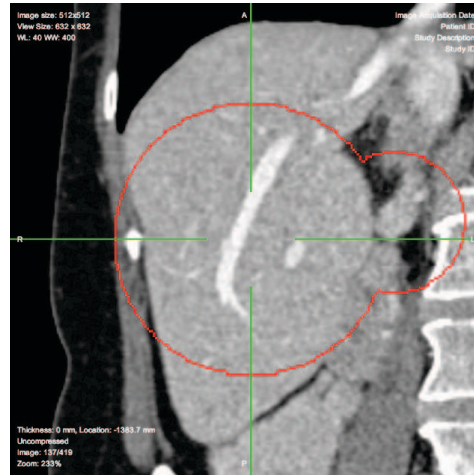
After a decision for the implementation of the 3D spheric cursor interface was made, some technical problems had to be solved.

The interface is composed of a circular yellow cursor that represents the projection of the 3D spheric cursor in a plane. Users can move this cursor from plane to plane and it should appear with the appropriate size for that plane zoom. It is also possible to change the cursor size (radius) using the mouse wheel. When the user clicks the mouse, a circular ROI is created and displayed in all planes the 3D spheric cursor is currently visible (Figure 6). This operation can be performed in any plane. If the user clicks and drags the mouse, the circular pattern is repeated forming a "painted" path. In this case, the 3D spheric cursor acts as a 3D brush and paint pixels in all image layers it touches. The result is a complex 3D shape that is shown by the interface as 3 projections in each plane, as shown in Figure 7. The interface can show the 3 planes at once or each one separately. Users can change which layer is shown in each plane and see the projection of the ROI in each layer. They can edit the ROI using any layer of any plane as a base to add or delete pixels from the ROI. After the 3D ROI is marked, the RadLex [8] terms to describe the lesion can be added using the existing ePAD image annotation mechanism. Finally, the 3D ROI will be stored in the ePAD server side as DICOM segmentation objects [2].

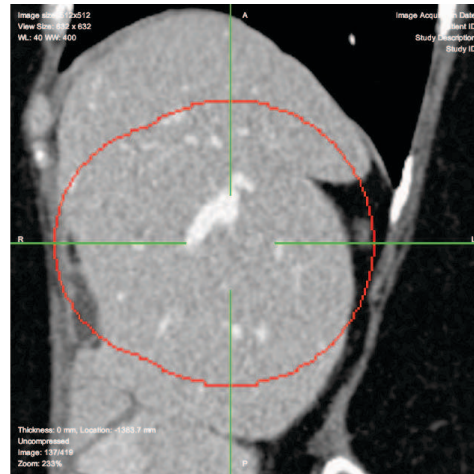
The technical problems for the interface implementation had to do with speed; Javascript was too slow to generate the cursor movements and ROI projections in real time. Again, WebGL



(a) Axial Plane



(b) Frontal Plane



(c) Sagittal Plane

Figure 7. **Complex 3D ROI** appearing as different projections on the 3 planes. Users can change plane layers to see the different projections in each layer.

was used to write the critical parts of the code to get the necessary execution speed.

#### IV. DISCUSSION

ePAD provides a promising approach to enable radiologists to perform measurements on target lesions and to produce quantitative imaging reports without substantially hampering the efficiency of their workflow. To date it has been limited in providing only 2D visualization and annotation of images.

To solve that, first the client ePAD viewer was rewritten as a framework capable of hosting multiple plugins. Then a plugin capable of showing 3D visualization of images, the ePAD 3D plugin, was developed. That development happened in four phases:

- 1) A 3D MPR viewer was developed (based on traditional desktop MPR viewers).
- 2) Three possible interfaces for 3D ROI marking were selected and video prototypes were created for each.
- 3) A study with radiologists was carried out using these interfaces. It determined that the preferred interface was the 3D Spheric Cursor.
- 4) The 3D Spheric Cursor interface was implemented as part of the plugin.

The current plugin implementation has some limitations. There is only one way to mark 3D ROIs (the 3D Spheric Cursor); it would be useful to have other methods. Also, there is no automation when marking ROIs, the process is totally manual.

#### V. CONCLUSION AND FUTURE WORK

We developed a freely available plugin for ePAD, the 3D plugin, which provides a major functionality to the tool: the ability to render volumetric image series in 3D, using the three MPR planes (axial, frontal and sagittal), and edit 3D ROIs. These ROIs are represented as DICOM segmentation objects to mark lesions. We also carried out a study to determine the best interface to provide a user for marking 3D ROIs, the 3D Spheric Cursor.

This 3D ROI capability will allow radiologist to mark lesions with much more precision than possible using 2D ROIs. It also has potential clinical impact in enabling physicians to efficiently marking regions on volumetric image datasets.

As future work, the plugin will let users retouch ROIs at the layer level, using a circle. This feature can be useful when correcting difficult spots. Another planned feature is to provide more automation to the process of marking ROIs. First, the spheric cursor will be used to mark lesions in a crude way; then the crude ROIs will be sent to the ePAD server and server-side algorithms (from server-side plugins) will fine tune the ROI; finally, the perfected ROI will be returned to the client for user inspection.

#### VI. ACKNOWLEDGMENTS

This work was partially funded by National Cancer Institute, National Institutes of Health, under grant U01CA142555. It was also partially funded by the National Council for Scientific and Technological Development (CNPq grant 481837/2008-6)

and the Coordination for the Improvement of Higher Education Personnel (CAPES).

#### REFERENCES

- [1] 3d slicer annotation image markup. Available at: [http://www.na-mic.org/Wiki/index.php/Projects:QIN:3D\\_Slicer\\_Annotation\\_Image\\_Markup](http://www.na-mic.org/Wiki/index.php/Projects:QIN:3D_Slicer_Annotation_Image_Markup). Accessed: 2015-02-08.
- [2] The digital imaging and communications in medicine (dicom) standard. Available at: <http://medical.nema.org/standard.html>. Accessed: 2015-02-09.
- [3] David S Channin, Pattanasak Mongkolwat, Vladimir Kleper, Kastubh Sepukar, and Daniel L Rubin. The cabig annotation and image markup project. *Journal of digital imaging*, 23(2):217–225, 2010.
- [4] EA1 Eisenhower, Patrick Therasse, Jan Bogaerts, LH Schwartz, D Sargent, Robert Ford, J Dancey, S Arbuck, S Gwyther, M Mooney, et al. New response evaluation criteria in solid tumours: revised recist guideline (version 1.1). *European journal of cancer*, 45(2):228–247, 2009.
- [5] O. Gevaert, J. Xu, C. D. Hoang, A. N. Leung, Y. Xu, A. Quon, D. L. Rubin, S. Napel, and S. K. Plevritis. Non-small cell lung cancer: identifying prognostic imaging biomarkers by leveraging public gene expression microarray data—methods and preliminary results. *Radiology*, 264(2):387–396, Aug 2012.
- [6] F. Gimenez, Jiajing Xu, Yi Liu, T.T. Liu, C.F. Beaulieu, D.L. Rubin, and S. Napel. On the feasibility of predicting radiological observations from computational imaging features of liver lesions in ct scans. In *Healthcare Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on*, pages 346–350, July 2011.
- [7] C. Klinger. Aim on clearcanvas workstation documentation. Available at: <https://wiki.nci.nih.gov/display/AIM/AIM+on+ClearCanvas+Workstation+Documentation>. Accessed: 2015-02-08.
- [8] Curtis P. Langlotz. Radlex: A new method for indexing online educational materials. *RadioGraphics*, 26(6):1595–1597, 2006. PMID: 17102038.
- [9] Mia A. Levy and Daniel L. Rubin. Current and future trends in imaging informatics for oncology. *Cancer journal (Sudbury, Mass.)*, 17(4):203–210, 2011.
- [10] Chris Marrin. WebGL specification. *Khronos WebGL Working Group*, 2011.
- [11] Peter Mildenerberger, Marco Eichelberg, and Eric Martin. Introduction to the dicom standard. *European radiology*, 12(4):920–927, 2002.
- [12] S. A. Napel, C. F. Beaulieu, C. Rodriguez, J. Cui, J. Xu, A. Gupta, D. Korenblum, H. Greenspan, Y. Ma, and D. L. Rubin. Automated retrieval of CT images of liver lesions on the basis of image similarity: method and preliminary results. *Radiology*, 256(1):243–252, Jul 2010.
- [13] D. L. Rubin, C. Rodriguez, P. Shah, and C. Beaulieu. iPad: Semantic annotation and markup of radiological images. In *AMIA Annu Symp Proc*, pages 626–630, 2008.
- [14] Daniel L. Rubin, Debra Willrett, Martin J. O'Connor, Cleber C. Hage, Camille Kurtz, and Dilvan A. Moreira. Automated tracking of quantitative assessments of tumor burden in clinical trials. *Translational Oncology (Online)*, 7:23–35, 2014.
- [15] Kleberson A. Serique, Alan Snyder, Debra Willrett, Daniel L. Rubin, and Dilvan A. Moreira. Using the semantic web and web apps to connect radiologists and oncologists. In *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, volume 1, pages 480–485. IEEE, jun 2012.
- [16] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [17] Stefan L. Zimmerman, Woojin Kim, and William W. Boonn. Informatics in radiology: Automated structured reporting of imaging findings using the aim standard and xml. *RadioGraphics*, 31(3):881–887, 2011.