X3D IN RADIATION THERAPY PROCEDURE PLANNING

Felix G. Hamza-Lup, Ivan Sopin, Dan Lipsa
School of Computing, Armstrong Atlantic State University, Savannah, Georgia, USA
felix@cs.armstrong.edu, dlipsa@cs.armstrong.edu

Omar Zeidan
M.D. Anderson Cancer Center Orlando, Florida, USA
Omar.Zeidan@orhs.org

Keywords: X3D, Web-based Simulation, 3D Modeling, Radiation Therapy.

Abstract: Radiation therapy, an increasingly available and effective cancer treatment solution, involves sophisticated machinery as well as careful planning. Interactive 3D simulations combined with accurate 3D patient specific data can improve the planning process saving time and resources in generating the optimal treatment plan. We illustrate the potential of X3D in radiation therapy, specifically radiation treatment planning. Embedding patient specific data (2D CT scans) in the interactive virtual setup improves the radiation therapy planning by early detection of collision cases. The X3D system may be used to support the decision making process as well as innovations in medical planning and training.

1 INTRODUCTION

While 3D simulation is just emerging as an accepted scientific discipline for medicine, the majority of applications are in the area of inter-operative navigation and training. We present the potential of X3D in radiation therapy, specifically radiation treatment planning.

Radiation therapy, an increasingly available and effective cancer treatment solution, involves sophisticated machinery as well as careful planning. An interactive X3D-based simulation combined with accurate 3D patient data obtained from CT scans and MRI will improve the planning process, saving time and resources in generating the optimal treatment plan. We illustrate the potential of X3D in radiation therapy, specifically radiation treatment planning. Embedding patient-specific data in the virtual setup improves the radiation therapy planning by early detection of collision cases.

A preliminary assessment leads us to believe that an accurate 3D virtual representation of the radiation therapy/surgery procedure supports the decision making process (e.g. specific configurations in the hardware for complex medical procedures) as well as innovation in medical planning and training.

In Section 2 we provide a brief introduction to the field of radiation therapy and associated hardware components. We also identify significant related work in radiation therapy simulation and training. Section 3 describes our solution and provides details of implementation. In Section 4 we focus on a very important component that allows us to bring patient-specific information inside the simulator to improve its realism and efficiency. In Section 5 we present the preliminary assessment implemented in collaboration with M. D. Anderson Cancer Center, Orlando.

2 BACKGROUND AND RELATED WORK

Radiation therapy is the careful use of high-energy radiation to treat cancer. A radiation oncologist may use radiation to cure cancer or for palliative purposes. About 50 to 60 percent of cancer patients are treated with radiation at some time during their disease (RadiologyInfo, 2006). Radiation destroys the ability of cancerous cells to reproduce, and the body naturally gets rid of these cells in time. A cancer patient may be treated with radiation alone (e.g. prostate cancer); however, sometimes radiation therapy is only a part of a patient's treatment. Patients can be treated with radiation therapy before surgery, enabling a less radical surgery than would otherwise be required. For example, some bladder cancer patients can keep their bladder if radiation therapy is effective.

A radiation oncologist may use the radiation generated by a machine outside the patient's body. This procedure is called External Beam Radiation Therapy. Radiation also may be given with radioactive sources that are injected in the patient; procedure called brachytherapy. Our contribution to treatment planning focuses on the External Beam Radiation Therapy (EBRT).

2.1 Time and Safety in the Treatment Process

EBRT involves advanced machinery that rotates around the patient, following a carefully planned schedule. The radiation delivery unit, called a linear accelerator (LINAC), consists of three main moving components: the gantry, the collimator, and the table/couch, as illustrated in Figure 1. The center of rotation of all these components is a virtual point in space called isocenter.

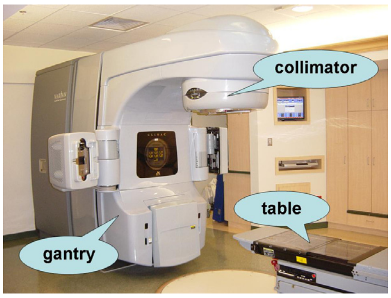


Figure 1: Linear accelerator with on-board imaging.

The principle of operation for such devices is fairly simple. Based upon the patient's tumor location, the hardware components change position and orientation while delivering pre-planned radiation doses to the patient on the table. However, in some cases, the complex relative orientations of all these components may cause a collision with the components themselves or between the components and the patient. In addition, LINAC head attachments and patient immobilization devices may also be a source of collision concerns. Therefore, a radiation treatment planner may generate a seemingly optimal plan, only to result in a collision when a "dry run" (i.e. an execution of the plan without radiation for collision checking) is performed on the LINAC. This results in a delay to the patient treatment, since the plan has to be revised to account for these unforeseen collisions. Additional time and resources must be invested to adjust the existing or create an alternative treatment plan.

Analytical methods for collision detection for LINAC-based radiation surgery have been proposed as a means to improve the EBRT planning process (Beange and Nisbet, 2006; Hua, Chang, and Yenice, 2004; Humm, Pizzuto, and Fleischman, 1995; Purdy, Harms, and Matthews, 1993). Analytical methods, even though accurate, are based on the hardware rotational and translational numerical values disregarding patient-specific geometry. Investigating previous work concerning graphical simulations of the LINAC system, we have identified the following limitations:

(1) sophisticated setups and additional software and hardware components;

(2) The simulations involve only generic patient body representations (Tsiakalos, Scherebmann, and Theodorou, 2001); hence collisions with patients are not accurately modelled or predicted;

(3) The simulations are local and can not be deployed via the web for potential collaboration with remote experts during treatment planning.

Researchers at the Hull Immersive Visualization Environment proposed a VR environment for training purposes (Beavis et al, 2006), using a virtual patient based on the visible human female dataset. Their prototype is targeted mainly at medical training. Our efforts are directed towards a distributed X3D-based system that will allow easy access from a web browser to the virtual room and will improve the actual planning process of the EBRT by providing a high-resolution model of all treatment components, including patient-specific geometry.

2.2 Web-based Systems in Medical Planning

Simulation-based Medical Planning has been recently investigated for cardiovascular disease (Steele et al, 2003). The Virtual Reality Modeling Language (VRML) has been employed to provide the visual web-based interface in the past. The European Institute of Telesurgery has proposed a 3D anatomical structure visualization and surgical planning system that allows manipulation and interaction on virtual organs extracted from CT-scan or MRI data (Chirstophe, Luc, and Jacques, 2002).

With the advent of the X3D standard and its extended functionality, the Internet-based systems for simulation gained momentum. Currently, the

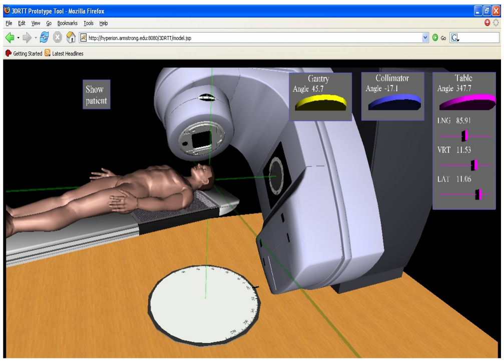
Medical Working Group, part of the Web3D consortium, is taking steps in developing an open interoperable standard for the representation of human anatomy based on input from a wide variety of imaging modalities. Our research and development efforts presented in the next section take into consideration and build on the existing technology, available in the public domain, allowing us to offer medical personnel free of charge access to the X3D simulation and associated paradigms.

3 AN X3D TOOL FOR COLLISION DETECTION

As we pointed out in Section 2.1, time-consuming "dry runs" are necessary to avoid collisions during the EBRT process. The "dry runs" do not take into consideration the patient's geometry. Depending on the size of the patient, unforeseen collisions might occur.

Providing a virtual 3D environment that will let the medical personnel view the execution of the treatment plan allows them to avoid potential collision cases and even experiment with new configurations. Our preliminary efforts in designing such an environment (Hamza-Lup, Davis, and Zeidan, 2006) provided a set of useful observations and guidelines for the current implementation.

The simulator implementation takes advantage of two technologies, Java and X3D. X3D (Web3D Consortium, 2006) is an ISO standard with an open architecture and a rich range of capabilities for real-time graphics processing that is employed in a wide array of domains and user applications. A successor to VRML, X3D is being developed by the Web3D Consortium as a refined standard (Lau et al, 2003).



3DRTT Prototype Tool - Mozilla Firefox

http://hyperion.armstrong.edu:8080/3DRTT/model.jsp

Gantry Angle 45.7

Collimator Angle -17.1

Table Angle 347.7

LNG 85.91

VRT 11.53

LAT 11.06

Figure 2: X3D Environment (3DRTT).

Besides X3D, in the development process we employ several software tools for 3D modeling. Figure 2 illustrates a snapshot of the virtual room (denoted 3D Radiation Therapy Treatment — 3DRTT) which models the real environment, depicted in Figure 1.

3.1 User Interaction

The simulator provides an intuitive floating graphical user interface (GUI) for controlling the angles and locations of the machine's parts (presented in Figure 2). The user may rearrange the GUI components to avoid important objects occlusion.

Volumetric slides and scrolls keep controlling operations simple and naturally fit in the 3D scene. The user can also show/hide a patient or the radiation beam generated by the collimator by turning designated switches on/off.

Besides the mouse manipulation capability, the user's viewpoint can be easily changed on the orthogonal axes. As a part of X3D, preset viewpoints provide handy control and fast overview potential and can be switched with a single keystroke.

An important component of the simulator is the ECMAScript (ECMA International, 1999), a JavaScript-like language that introduces additional functionality into X3D. The ECMAScript functions provide the GUI's interactivity. For instance, a user shows or hides the patient or the beam by clicking the appropriate buttons, or the values change when the user rotates the gantry either by dragging it or scrolling the associated control element. The position of the object on the screen is synchronized with the position of the corresponding scroll wheel.

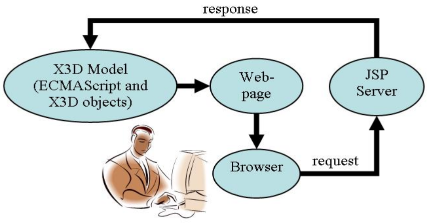


Figure 3: Client-Server Interaction.

The X3D model containing the ECMAScript and the optimized polygonal objects is embedded in a web page that the user accesses from a local machine. When the user submits new values on the web page, a Java servlet processes them. The server does not only generate a new web page, but also deploys a new X3D model, as described in Figure 3. Every user of the simulator is assigned a unique session ID. Thus, clients using the same model avoid conflicts in controlling the simulator. So, 3DRTT has the distributed functionality of a 3D distributed visual system on the web and can easily be shared between researchers and remotely located medical personnel.

3.2 3D Models Acquisition and Processing

A 3D laser scanner from Faro™ Technologies is employed to collect point clouds from several viewpoints. Once the point clouds are collected, they are merged into one cloud based on a set of specially designated markers. We filter the noise and wrap the valid points into a polygonal model.

To improve the rendering process, we decimate the model by removing redundant polygons in flat areas while securing a sufficient number of polygons in the regions with complex geometry. The polygonal model is exported into an X3D object and employed as a part of the scene. All filtering, wrapping, smoothing, and exporting operations are performed using the built-in algorithms in the commercially available Raindrop GeoMagic™ software.

Considering the geometrical complexity and the high-polygonal resolution of the model, we have to optimize the polygonal model such that adequate frame-rates (25 FPS or more) are obtained on machines with less rendering power. To improve the rendering speed and reduce the file size, we are using textures, simulating the geometry of complex areas. Textures also save development time because instead of processing a complex region we rather substitute it with a texture that does not require a long time to generate. Another positive side effect is the network traffic reduction.

3.3 Collision Detection — Special Collision Situations

Currently, the X3D standard supports only avatar collision detection, meaning that users may "collide" with virtual objects as they travel through the scene.

Only a few research groups attempted to implement the object-to-object collision detection directly in a VRML or X3D environment. For instance, the V-COLLIDE library (Hudson et al, 1997) provides fast collision detection for arbitrary polygonal objects based on the Robust and Accurate Polygon Interference Detection (RAPID) algorithm (Gottschalk, Lin, and Manocha, 1996) and is potentially capable of computing collisions in VRML. However, to the best of our knowledge, none of these collision methods have been fully implemented and tested in a VRML/X3D player.

Besides the visual check for collisions in 3DRTT, we are currently developing an automatic collision detection system via bounding boxes that approximates the shapes of virtual objects.

4 PATIENT 3D MODELS FROM CT DATA

An important issue we address with this work is real-patient data embedded in the X3D simulation. We have to efficiently convert a set of CT scans of a patient to a polygonal model of the patient's body. The set of CT scans used is stored using Digital Imaging and Communication in Medicine (DICOM) standard.

We process CT scans using the Visualization Toolkit (VTK) (Schroeder, Martin, and Lorensen, 1996), a software system for 3D computer graphics, image processing, and visualization. The first step in our processing is to select from CT scans a volume that eliminates the table on which the patient is positioned. This operation reduces the number of polygons obtained in our 3D patient model.

We apply the marching cubes algorithm (Lorensen and Cline, 1987) with a value that selects the isosurface of the patient's skin. We used the same value for the skin as it was done in (Lorensen, 2006). The marching cubes algorithm checks if the corner values of a cube are above or below the isosurface value. If some corner values are above and some are below, then the isosurface obviously intersects the cube, and we can compute the intersection polygon by using a lookup table. The lookup table provides a fast way to find the edges intersected by the isosurface based on what corners values are above and below the isosurface.

The application of the marching cubes algorithm produces a polygonal model that includes the skin of the patient as well as internal organs. We have applied this algorithm on a set of 133 CT scans of a patient torso. The number of polygons obtained (approximately 500,000) significantly slowed down the interaction with our web-based application. To reduce the number of polygons, we apply a decimation operation (Schroeder, Zarge, and Lorensen, 1992) which reduces the number of polygons to approximately 100,000. Decimation works by evaluating the distance between each point and the average plane of the triangles using the point. If that distance is small, the point and all triangles using it can be deleted. Obviously, the "hole" left has to be re-triangulated. Once re-triangulation is accomplished, we apply a Laplace smoothing operation (Field, 1988) that modifies the position of each vertex to be the average of the neighbouring vertices.

To obtain a "clean" polygonal model that contains only the skin of the patient (remember, we are interested in visual collision checks between the hardware and the patient), we apply a polygon connectivity filter that further reduces the number of triangle strips to about 3,000. A challenge in applying this algorithm is specifying the connected surface. Choosing the maximum connected surface isolates the skin of a patient in most cases. The result is illustrated in Figure 4.

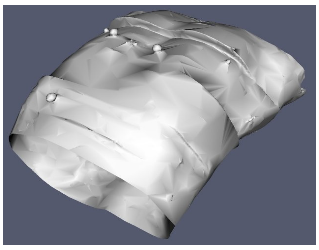
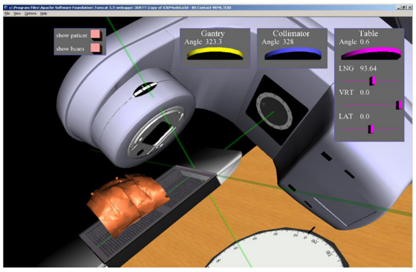


Figure 4: 3D Torso Model from CT Scans.

A final step in the process is the conversion of the 3D model into an X3D object. The X3D object is embedded in the scene on the LINAC table in the same position and orientation as the real patient (Figure 5).



c:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\3DRTT\Copy of X3DModel.x3d - BS Contact VRML/X3D

Gantry Angle 323.3

Collimator Angle 328

Table Angle 0.6

LNG 93.64

VRT 0.0

LAT 0.0

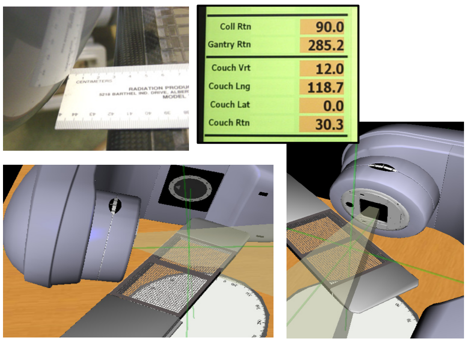
Figure 5: Patient-specific 3D model on the Web.

Now, the medical personnel can virtually execute the treatment plan and check for collisions with the patient body. Of course, the CT data is limited to a specific body part (in this case the torso). We are planning an improvement to this approach, i.e. to generate a full body patient model from incomplete CT scan data. We think that such an improvement will help detect potential collisions with other patient body parts, e.g. hands, legs, etc.

5 PRELIMINARY ASSESSMENT

We have deployed the system on a secure web site and allowed medical personnel from M.D. Anderson Cancer Center, Orlando, to remotely access the web-based simulation environment. Their first subjective reaction was that the X3D world is very realistic and that it improves their confidence in running the LINAC in a real scenario, since they can now visualize the relative position of all the components involved, including the patient.

To objectively test collision scenarios, we asked a radiation therapy technician and a therapist to simulate a plan that contains collisions among the system components (illustrated in Figure 6). The result was that the X3D simulation provides an accurate representation of the LINAC (specifically, the Varian 23iX LINAC) that can predict any collision scenarios with centimeter accuracy.



Coll Rtn 90.0

Gantry Rtn 285.2

Couch Vrt 12.0

Couch Lng 118.7

Couch Lat 0.0

Couch Rtn 30.3

Figure 6: Visual Collision Validation.

The preliminary assessment using visual inspection from different angles, illustrated in Figure 6, provides an early validation for the accuracy of the simulator. The tool can be used off-line by planners to inspect their patient-specific beam arrangements before simulation runs. We are in the process of measuring the impact of the X3D simulator on the EBRT planning and execution efficiency.

6 CONCLUSIONS AND NEAR FUTURE

We have presented the development of an X3D Web-based simulator for External Beam Radiation Therapy treatment simulation. The ability to detect/predict a possible collision between all LINAC components for a given patient eliminates the need for backup plans and saves planning time.

In addition, it enables the planner to explore different and unconventional gantry-couch-collimator combinations for treatment that may give rise to better quality plans.

We are developing a database, containing X3D objects corresponding to various EBRT equipment models, to meet the demands of a broader range of medical facilities and applications.

We hope that our research and development efforts will advance the quality of the radiation therapy process and consequently will improve cancer patients' treatment and recovery.

ACKNOWLEDGEMENTS

Support for this work came from M. D. Anderson Cancer Foundation. We thank Anne Ocheltree and Chris Morabito from FARO™ Technologies Orlando for their support in obtaining the high resolution 3D models for the EBRT room.

REFERENCES

Beange, I., Nisbet, A., 2000. A collision prevention software tool for complex three-dimensional isocentric set-ups. In *British Journal of Radiology*, vol. 73, pp. 537-541.

Beavis, A., Ward, J., Bridge, P., Appleyard, R., Phillips, R., 2006. An Immersive Virtual Environment for Training of Radiotherapy Students and Developing Clinical Experience. In *American Association of Physicists in Medicine*, vol. 33, pp. 2164-2165.

Chirstophe, K., Luc, S., Jacques, M., 2002. PACS-based interface for 3D anatomical structure visualization and surgical planning. In *Proc. SPIE — Medical Imaging 2002: Visualization, Image-Guided Procedures and Display*, vol. 4681, pp. 17-24.

ECMA International, 1999. ECMAScript Language Specification. In *Standard ECMA-262*, vol. 2006, 3 ed. Geneva.

Field, D. A., 1988. Laplacian smoothing and Delaunay triangulations. In *Communications in Applied Numerical Methods*, vol. 4, pp. 709-712.

Gottschalk, S., Lin, M., Manocha, D., 1996. OBB-Tree: A Hierarchical Structure for Rapid Interference Detection. In *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, New Orleans, Louisiana.

Hamza-Lup, F., Davis, L., Zeidan, O., 2006. Web-based 3D Planning Tool for Radiation-Therapy Treatment. In *Web3D 2006: 11th International Symposium on 3D Web Technology*, Columbia, Maryland.

Hua, C., Chang, J., Yenice, K., 2004. A practical approach to prevent gantry-couch collision for linac-based radiosurgery. In *Medical Physics*, vol. 31, pp. 2128-2134.

Hudson, T., Lin, M., Cohen, J., Gottschalk, S., Manocha, D., 1997. V-COLLIDE: Accelerated Collision Detection for VRML. In *Proceedings of the Second Symposium on the Virtual Reality Modeling Language*, Monterey, California.

Humm, J., Pizzuto, D., Fleischman, E., 1995. Collision detection and avoidance during treatment planning. In *International Journal of Radiation Oncology, Biology and Physics*, vol. 33, pp. 1101-1108.

Lau, R., Li, F., Kunii, T., Guo, B., Zhang, B., Magnenat-Thalmann, N., Kshirsagar, S., Thalman, D., Gutierrez, M., 2003. Emerging Web Graphics Standards and Technologies — Web Graphics Tutorial. In *Computer Graphics and Applications*, vol. 23, pp. 2-11.

Lorensen, B., 2006. *Make Your Own Visible Woman*, [Online], Available: http://www.crd.ge.com/esl/cgsp/projects/makevw [9 Jan 2007].

Lorensen, W. E., Cline, H. E., 1987. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, Anaheim, California.

Purdy, J. A., Harms, W. B., Matthews, J. W., 1993. Advances in 3-dimensional radiation treatment planning systems: room-view display with real time interactivity. *International Journal of Radiation Oncology, Biology and Physics*, vol. 27, pp. 933-944.

RadiologyInfo, 2006. *Introduction to Radiation Therapy (Radiation Oncology)*, [Online], Available: http://www.radiologyinfo.org/en/info.cfm?pg=intro_onco&bhcp=1 [10 Jan 2007].

Schroeder, W. J., Zarge, J. A., Lorensen, W. E., 1992. Decimation of triangle meshes. In *SIGGRAPH '92: Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, Chicago, Illinois.

Schroeder, W. J., Martin, K. M., Lorensen, W. E., 1996. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, San Francisco, California.

Steele, B., Draney, M., Ku, J., Taylor, C., 2003. Internet-Based System for Simulation-Based Medical Planning for Cardiovascular Disease. In *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, pp. 123-129.

Tsiakalos, M. F., Scherebmann, E., Theodorou, K., 2001. Graphical treatment simulation and automated collision detection for conformal and stereotactic radiotherapy treatment planning. In *Medical Physics* vol. 28, pp. 1359-1363.

Web3D Consortium, 2006. *What is X3D?* [Online], Available: http://www.web3d.org/about/overview/ [10 Jan 2007].