An open-source platform for interactive collision prevention in photon and particle beam therapy treatment planning

F Hueso-González, P Wohlfahrt, D Craft and K Remillard
Department of Radiation Oncology, Massachusetts General Hospital and Harvard Medical School, Boston, MA 02114, United States of America
E-mail: fernando.hueso@uv.es

Keywords: radiotherapy, collision, treatment planning

Abstract
We present an open-source platform to aid medical dosimetrists in preventing collisions between gantry head and patient or couch during photon or particle beam therapy treatment planning. This generic framework uses the native scripting interface of the particular planning software to import STL files of the treatment machine elements. These are visualized in 3D together with the contoured or scanned patient surface. A graphical dialog with sliders allows the interactive rotation of the gantry and couch, with real-time feedback. To prevent a future replanning, treatment planners can assess in advance and exclude beam angles resulting in a potential risk of collision. The software platform is publicly available on GitHub and has been validated for RayStation with actual patient plans. Furthermore, the incorporation of the complete patient geometry was tested with a 3D surface scan of a full-body phantom performed with a handheld smartphone. With this study, we aim at minimizing the risk of replanning due to collisions and thus of treatment delays and unscheduled consumption of manpower. The clinical workflow can be streamlined at no cost already at the treatment planning stage. By ensuring a real-time verification of the plan feasibility, the script might boost the use of optimal couch angles that a planner might shy away from otherwise.

1. Introduction

Treatment of cancer patients with accelerated charged particles or photon beams is performed ideally from various incidence angles [1] to better spare normal tissue or Organs At Risk (OAR) surrounding the tumor. To enable the irradiation from any direction out of a 4π sphere, the treatment head is mounted on a rotating gantry, whereas the patient couch can rotate around a vertical axis, in addition to three-dimensional (3D) translations. In the case of particle beam therapy, the treatment head (also known as nozzle) may comprise a moving snout that supports apertures, compensators and range shifters, that are positioned close to the patient surface.

As a consequence of the dynamically moving gantry head, snout and couch, there is a risk of damage of equipment, treatment interruption or even patient injury. To ensure the overall safety, aside from emergency buttons, surveillance cameras [2] and touch guard fins [3], potential collisions between gantry head and couch or patient need to be assessed in advance and prevented [4]. Throughout the last three decades, different approaches have been developed to aid treatment planners in the avoidance of irradiation angles with risk of collision. These were based on simplified analytical calculations [5–11], graphical simulations [12–23] or experimental reference measurements [24–31]. In some cases, the combination of treatment parameters leading to a collision are depicted as keep-out areas in a set of reference charts, or implemented as a warning feedback within the treatment software. In others, the user can move the isocenter and rotate the gantry interactively, and the risk of collision is detected automatically or assessed visually.

Collision detection during treatment planning is one important tool in the context of personalized medicine, where the optimum treatment plans for every patient are sought, but must be feasible at the same time. Despite extensive research and the
an effective collision detection work be combined coherently in the same platform ensuring couch for delivery. The respective information should

Massachusetts General Hospital radiotherapy centers. In many cases, radiotherapy decades, there is no standardized solution applied in

the capabilities and conformality of the patient-specific from introducing couch angles and non-coplanar

Dosimetric perspective. In cases where the treatment machine showing the beam incidence. Some users have privately developed basic collision detection scripts in RayStation by modeling the gantry geometry as combinations of boxes and cylinders [21, 28, 29].

RadCollision, an open-source platform for collision assessment in TPS that:

- Is licensed under GPLv3 [35] at no cost, and can be downloaded online,
- Is maintained by the scientific and clinical community through public repositories,
- Can progressively support TPS from further vendors,
- Is easily adaptable by any institution,
- Does not require purchasing additional hardware,
- Does not need expert knowledge about software,
- Is embedded in every TPS and does not require external software or data transfer to other servers,
- Is patient-specific,
- Provides a realistic 3D visualization of nozzle, couch and patient, rather than reference charts,
- Is modular, so that further room elements can be added into (or removed from) the visualization by the end user,
- Depends on 3D model input files in StereoLithography (STL) format,
- Aids treatment planners in choosing beam angles with interactive sliders,
biomedical procedures and workflow are presented in sections 4 and 5.

Section 3. A brief discussion and the main conclusions for the RayStation scripting interface is validated in

Optionally incorporates the full patient geometry

Relies on an initial 3D modeling of the treatment machine, or the willingness of vendors to provide their 3D models to hospitals,

Optionally incorporates the full patient geometry recorded with any 3D scanner or surface imaging device.

This manuscript is organized as follows. The software framework and details of its implementation are discussed in section 2. The application of the platform for the RayStation scripting interface is validated in section 3. A brief discussion and the main conclusions of the paper are presented in sections 4 and 5.

2. Materials and methods

2.1. Software architecture and system description

The proposed software model for collision prevention is illustrated in figure 1. It is designed to be as embedded as possible into the TPS used by the dosimetrist, but keeping the flexibility and modularity, as specified in section 1.

First, it is assumed that the 3D outer surfaces of each treatment machine and any other room elements relevant for collision are available to the hospital. These may be requested to the vendors upon purchase or acquired later under an Non-Disclosure Agreement (NDA). Or they can be downloaded from online 3D stores or community repositories. Alternatively, one may generate these in situ based on a 3D scan of the machine, or experimental measurements [15, 23, 34]. Regardless of the source, the 3D model shall be centered at the room isocenter, processed to remove unnecessary internal sub-parts, and exported as STL format [36], one for each subpart of the machine moving independently. These files are stored in a shared directory of the hospital servers.

Second, the model of the patient relies on the external contour of the CT or magnetic resonance imaging (MRI) dataset of the patient, which is usually already available as a region of interest (ROI) within the TPS and thus no specific action is needed. If a more complete model of the patient is required, phantom-based extensions [10], in-room 3D cameras [30] or even scans from handheld devices, see section 2.3, may be used. These 3D scans need to be converted to STL format with e.g. the Meshlab open-source software [37] and imported into the TPS as external contour.

Third, it is required that the deployed TPS software comprises an embedded viewer of ROIs as 3D surfaces and provides a scripting interface with multithread support. Three public methods are essential to support this application: the ability to import an STL file as an ROI, to transform (rotate and translate) any ROI with a $4 \times 4$ matrix, and to calculate the region of overlap between two ROIs.

Considering this set of prerequisites, we propose and design a new open-source software online platform, named RadCollision, as a generic tool for collision prevention in radiotherapy. It is divided in a core layer and an interface layer, whereas the setup layer lies outside of the public platform.

Its core layer defines the abstract classes and methods. For example, an element rotating around isocenter, like the gantry head, or any object translating in 3D and rotating around the vertical axis, like the couch. This layer also generates the corresponding transformation matrices depending on the irradiation angle or couch position according to the DICOM (IEC 61 217) coordinate system conventions [38].

The interface layer handles the graphical user interface (GUI), as well as the communication with the application programming interface (API) functions of the specific TPS. Because the function signatures might differ, and each TPS may support a different programming language within their scripting interface, this layer may have to be duplicated and specialized for every case (see shadow in figure 1), wrapping the calls to the generic core methods.

The setup layer is hospital-specific and consists of a database of all 3D models of the available machines and any other relevant room elements. Each part has to be assigned to one of the abstract classes defined in the core layer according to its particular motion behavior (degrees of freedom).

During treatment planning, once the patient is contoured, the user can start the collision prevention software. The program automatically chooses the machine and couch model from the active treatment plan among those available in the hospital database (setup layer). The selected ones are loaded as ROIs by the TPS (scripting interface). Then, the GUI dialog (interface layer) allows for the adjustment of irradiation settings (gantry angle, couch angle, snout extraction, etc). The software transforms in real-time the ROIs corresponding to the treatment machine, and calculates any collision (overlap of ROIs) with the patient or couch in the background. There is also a GUI button to automatically calculate the collision report for each beam defined in the treatment plan.

For dynamic arcs, the collision is calculated in steps of one degree.
2.2. Implementation for RayStation

We exemplarily validated the proposed software model for the TPS RayStation. The interface layer was written in IronPython [39], the original implementation language of the scripting library of RayStation. The threaded GUI relies on the native WinForms library [40]. The script is publicly available on the MGH radiation oncology GitHub organization [41], and requires the use of RayStation 8B or newer versions.

The main three functions from the RayStation API called by the interface layer are:

(i) ImportRoiGeometryFromSTL
   (FileName, TransformationMatrix)

(ii) TransformROI3D
     (TransformationMatrix)

(iii) ComparisonOfRoiGeometries(RoiA, RoiB, ComputeDistanceToAgreement Measures)

The import of the STL file (function 1) is done only once, at script startup. This function is available since RayStation version 8B. Each time a slider of the GUI is changed, the 3D transformation (function 2) has to be applied on the already imported ROI. This $4 \times 4$ affine transform matrix, specifically defined for the treatment isocenter, is computed independently for each sub-part of the couch or nozzle according to the motion behavior initially configured by the user (setup layer). If automatic collision detection is enabled in the GUI, the third function calculates if two ROIs overlap via the dice similarity coefficient (DSC) [42].

It shall be noted that, as the 3D modeling in RayStation is done in the patient coordinate system, the simulation of couch angles is done by rotating the room elements (gantry and optionally walls) in the opposite direction rather than by rotating the couch model.

2.3. Optional 3D surface scan

The presented framework, cf. figure 1, is compatible with the import of a 3D surface scan of the full patient geometry, in order to detect potential collisions with parts of the body outside the field of view of the CT scan. Except for the requirement to export the 3D surface scan as STL file, no prior assumptions are
needed for the scanning device. Finally, the 3D scans have to be rigidly registered to the CT scan geometry.

To illustrate this workflow, we acquired a CT scan of an anthropomorphic female phantom (Alderson Research Laboratories, Stanford, CT, USA) using a GE Discovery RT CT scanner (GE Healthcare, Chicago, IL., USA), which served as ground-truth geometry. Subsequently, a 3D surface scan with a handheld iPhone XS (Apple, Cupertino, CA, USA) was performed. The front face camera of the smartphone comprises depth sensor technology [43], that can be used in combination with the free application Capture: 3D Scan Anything (Standard Cyborg, Inc, San Francisco, CA, USA) to obtain a 3D surface scan of an object in Polygon File Format (PLY) format. The PLY file can be imported into e.g. Slicer3D for rigid registration with the CT scan geometry, and the resulting mesh can be exported as STL file (or even directly as contour in an RT structure file). A similar procedure can be conducted with any other 3D scanner type.

The conformity of external contours derived from the CT scan and the 3D surface scan was assessed by the minimal contour displacement and Hausdorff distance, defined as the 95th quantile of absolute contour distances for each axial CT slice from head to pelvis of the anthropomorphic female phantom [44].

3. Results

The implementation of RadCollision for the RayStation TPS was evaluated qualitatively with actual patient plans from the MGH radiotherapy department. Four patient plans, which were found infeasible during collision check by the therapists in the past, were analyzed retrospectively. Based on the 3D visualization and the collision report results, cf. figure 2, collisions with the couch were found at similar angles than those reported experimentally (within 2 degrees). The replanned treatments (with other beam angles or isocenter positions) were also studied, showing no effective collision for the selected incidence directions. In a fifth case, the simulation was applied prospectively, before patient treatment, and the predicted absence of collisions was confirmed during a dry run with the patient in position. A quantitative analysis of the overall prediction accuracy was not performed, as it depends on the input data (more details in section 4) rather than on the proposed software model. Hence, the latter has been the focus of this manuscript.

In figure 3, the software was tested with the model of a proton treatment room and a robotic system for patient positioning consisting of two articulated arms The robot configuration was automatically calculated based on the couch position, in order to assess the collision risk between the robot arms and the nozzle. The interactivity capabilities of the GUI are shown in figure S1 (available online at stacks.iop.org/BPEX/6/055013/mmedia) for photon therapy (top) and proton therapy (bottom).

The quantitative analysis of the accuracy of the 3D surface scan (figure S2) with respect to the ground-truth geometry derived from a CT scan is shown in figure 4. The geometry obtained by the 3D surface scan is in general slightly larger than the CT geometry, which provides more conservative results for the collision test. The median distance between the two external contours in the evaluation area, excluding the
region of contact between patient and couch surface, is roughly 1.6 mm. In 8% of all cases, the contour pixels from the 3D surface scan are inside of the external contour determined on the CT scan (negative minimal contour displacement) with a mean absolute deviation of $(−1.6 \pm 1.0)$ mm. Overall, the minimal contour displacement was within $−1.7$ mm (2.5th quantile) and $5.9$ mm (97.5th quantile) at a 95% confidence level. Differences larger than 10 mm were occasionally observed for some CT slices, in particular in the neck.
region, which were mainly caused by a non-optimal orientation of the smartphone during the 3D surface scanning test.

4. Discussion

This manuscript proposes RadCollision, a potential generic solution for collision assessment in a variety of treatment modalities by importing STL files of the machine and room elements through the scripting interface of a TPS, cf. figure 1. This approach is as embedded as possible in the workflow of dosimetrists, open-source, modular, and does not imply any investment for the hospital.

However, this framework requires some coordinated initial efforts from several parties. First, the different TPS softwares have to support the import of STL files as ROI through their scripting interface, cf. figure 1, as well as to enable a 3D viewer tool. To date, we are only aware of Slicer3D and RayStation TPSs to offer these functionalities. Second, to obtain the highest precision, the treatment machine vendors have to provide the 3D models to the hospitals under an NDA. Third, hospital staff has to process and organize these models into a database. Other collision detection methods published in the literature are more specific and sophisticated than the presented solution, but also more complex to implement and require the acquisition of further hardware like fixed cameras or room lasers, and potentially the use of external proprietary software [21] and the need of data transfer from the TPS. This might be an obstacle for implementation in a widespread context. In contrast, the RadCollision framework is embedded (provided a set of prerequisites), modular and scalable, by allowing through the setup layer (figure 1) the progressive addition of other sub-elements of the treatment room like electron applicators or imaging detector panels [45], without the need of upgrading the TPS software. It also allows for (but does not force to) the incorporation of the complete 3D patient surface, and is agnostic about the 3D scanning source, e.g. a handheld smartphone (figure S2), as long as the output is converted to STL format.

It should be noted that the reliability of the collision assessment within this software platform depends on the accuracy of the input data, cf. figure 1, rather than on the software itself (numeric rounding aside). The following sources of uncertainty are identified:

- 3D models of the machine and couch
- 3D scan (if performed)
- Patient positioning (treatment versus imaging)
- Patient anatomy (variations over the treatment course)
- Patient motion (variations within a treatment fraction)

In general, 3D models of the machine elements provided by the vendors are very precise (manufacturing tolerance and specifications), whereas the patient representation has a higher error, either due to the restricted field of view of the CT scan, or due to the inaccuracy of the 3D scan of the patient surface, cf. figure 4.

To use this tool for patients, reasonable safety margins for collisions should be introduced by inflating the external ROI in the TPS. The overall collision prediction accuracy, i.e. True Positive (TP) rate, will be a combination of the aforementioned uncertainties and the chosen safety margin. The specific choice should be in accordance with the estimated magnitude of these errors, that might be specific for each machine, 3D scanner, patient age (motion) and tumor site. In a clinical setting, a way to calibrate the safety margin could be to draw the Receiver Operating Characteristic (ROC) curve of the collision prediction in an initial study for different margins and cases, and find a compromise between maximizing the sensitivity and minimizing the False Positive (FP) rate.

This generic software paradigm was realized for the TPS RayStation, using IronPython as scripting language. A GUI dialog with sliders (figure 2) allows for an interactive adjustment of beam angle, couch angle and snout extraction with real-time feedback, and reports the risk of collision. The automatic collision detection runs on a separate thread pool, not to freeze the feedback of the GUI, cf. figure S1. Nonetheless, it can be switched off by the user for reducing the overall server load [20], if needed. A GUI button triggers the calculation of the collision report for all the beams and arcs in the treatment plan. The code is openly available in a GitHub repository [41] and can be maintained by the collaborative efforts of the scientific and clinical communities.

The integration of this tool in the clinical routine of a radiotherapy department might contribute to an overall improvement of the daily workflow: less or no time is required for actual collision checks using the treatment machine, thus reducing the workload of therapists as well as the machine time not available for patient treatments. Moreover, delays in the beginning of patient therapy are prevented, which otherwise emerge when a collision is found during the dry run at the treatment room, requiring unscheduled allocations of time and resources for replanning.

Furthermore, the 3D visualization of the actual treatment room at the planning stage facilitates the
selection of optimal beam and couch angles, which can in turn improve the dosimetric quality of the plan. By providing a real-time assurance that the selected angles do not present a risk of collision, i.e. a risk of cost-intensive replanning, the dosimetrists are less likely to shy away from irradiation geometries beneficial from the dose perspective. In this regard, the script could be most helpful for clinical cases such as stereotactic treatments, extremities, partial breast irradiation and prone breast treatments, electron beams, as well as plans with drastically anterior or posterior isocenters. The presented tool is expected to aid in the development of optimally individualized treatment plans.

In the future, users of other TPS software might contribute to the public repository writing their specific interface layer, cf. figure 1, and request their vendors to support this software model in their future releases. Namely, a 3D viewer tab and three specific functions would need to be implemented on their side: the ability to import a 3D model STL file as an ROI, the transformation of ROIs based on an affine transform matrix, and the calculation of the overlap between two contours.

5. Conclusions

An open-source software architecture for patient-specific collision assessment in external beam radiotherapy is proposed. It relies on the native scripting interface of each TPS, and requires its ability to import STL files of the patient couch and treatment head as ROIs. These are superimposed with the contoured patient geometry in the 3D visualization tab. It also enables the incorporation of the complete patient geometry, that might not be fully represented in the underlying CT scan, based on any 3D surface scanning device. This can aid the planner in estimating whether the treatment head will collide with any part of the patient, for example with the arms of breast patients. Hence, it minimizes the risk of replanning and thus of treatment delays, and allows for the choice of optimum and feasible irradiation angles.

The presented collision detection tool was evaluated for the RayStation TPS with actual patient plans, with no additional external software required. It will be included as part of the clinical workflow of dosimetrist of the radiotherapy section of MGH as soon as an upgrade to RayStation version 8B (or higher) for clinical use is performed. Future work will be devoted to the inclusion of the 3D models of accessories like wedges and bolus, and to the automatic feasibility assessment of final beam angles and dynamic arcs as a prerequisite for the treatment plan approval.

Acknowledgments

We thank I Andras, T Borntfeld, D M Edmunds, D Gierga, D Kunath, R Löschner, M Luzarra, G Sharp, J Smeets, J Söderberg, M Spiegel, K Stützer, J M. Verburg, E Vidholm, S Yan and W Zou for scientific advice and discussions, and the Lunder team at MGH for technical support. This work was supported in part by the Federal Share of program income earned by Massachusetts General Hospital on C06-CA059267, Proton Therapy Research and Treatment Center.

The authors have no relevant conflicts of interest to disclose.

ORCID iDs

F Hueso-González  @ https://orcid.org/0000-0003-3217-0090
P Wohlforth  @ https://orcid.org/0000-0002-2121-0934
D Craft  @ https://orcid.org/0000-0003-3093-718X

References

[1] Bortfeld T and Schlegel W 1993 Optimization of beam orientations in radiation therapy: some theoretical considerations Phys Med Biol. 38 291–304
[2] Nguyen S M, Chlebik A A, Okh A J and Wong K K 2019 Collision risk mitigation of Varian TrueBeam linear accelerator with supplemental live-view cameras Pract Radiat Oncol. 9 e103–9
[3] Tsai J S et al 1991 Quality assurance in stereotactic radiosurgery using a standard linear accelerator Int J Radiat Oncol Biol Phys. 21 737–48
[4] Hoopes D J, Ford E C, Ezzell G A, Dicker A P, Chera B S and Potters L 2017 Incident learning for stereotactic radiation therapy from RO-ILS: radiation oncology incident learning system Int J Radiat Oncol Biol Phys. Supplement, Proc of the ASTRO. 99 S46–7
[5] Yorke E D 1989 The geometry of avoiding beam intersections and blocking tray collisions Med Phys. 16 288–91
[6] Humm J L 1994 Collision avoidance in computer optimized treatment planning Med Phys. 21 1053–64
[7] Muthuswamy M S and Lam K L 1999 A method of beam-couch intersection detection Med Phys. 26 229–35
[8] Beange I and Nisbet A 2000 A collision prevention software tool for complex three-dimensional isocentric set-ups Br J Radiol. 73 537–41
[9] Furhant E E et al 2002 Clearance assurance for stereotactic radiosurgery and radiotherapy Med Phys. 29 45–50
[10] Noutsokou E, Bedford J L and Webb S 2003 Patient-specific planning for prevention of mechanical collisions during radiotherapy Phys Med Biol. 48 N313–21
[11] Hua C, Chang J, Venice K, Chan M and Amols H 2004 A practical approach to prevent gantry-couch collision for linac-based radiosurgery Med Phys. 31 2128–34
[12] Sherouse G W and Chaney E L 1991 The portable virtual simulator Int J Radiat Oncol Biol Phys. 21 475–82
[13] Kessler M L, McShan D L and Fraass B A 1995 A computer-controlled conformal radiotherapy system. III: Graphical simulation and monitoring of treatment delivery Int J Radiat Oncol Biol Phys. 33 1173–80

8
Bayouth L E, Muthuswamy M, Kalend A M, Izadbakhsh M M and Greenberger J S 1995 94 Computer controlled conformal therapy-analytical and computer simulation of collision avoidance and treatment delivery Int J Radiat Oncol Biol Phys. 32 188

Tsikalos M F, Schrebmman E, Theodorou K and Kappas C 2001 Graphical treatment simulation and automated collision detection for conformal and stereotactic radiotherapy treatment planning Med Phys. 28 1359–63

Ward J et al 2007 Immersive visualization with automated collision detection for radiotherapy treatment planning Stud Health Technol Inform. 125 491–6 (https://pubmed.ncbi.nlm.nih.gov/17377334/)

Hamza-Lup F, Sopin I and Zeidan O 2007 Comprehensive 3D virtual visualization for radiotherapy radiation planning Stud Health Technol Inform. 125 164–6

Hamza-Lup F G, Sopin I and Zeidan O 2008 Online external beam radiation treatment simulator Int J Comput Assist Radiol Surg 3 275–81

Glaser S, Warfel B, Price J, Sinacore J and Albuquerque K 2012 Effectiveness of virtual reality simulation software in radiotherapy treatment planning involving non-co-planar beams with partial breast irradiation as a model Technol Cancer Res Treat. 11 409–14

Zou W et al 2012 A clinically feasible method for the detection of potential collision in proton therapy Med Phys. 39 7094–101

Yu V Y et al 2015 The development and verification of a highly accurate collision prediction model for automated noncoplanar plan delivery Med Phys. 42 6457–67

Suriyakumar V M, Xu R, Pinter C and Fichtinger G 2017 Open-source software for collision detection in external beam radiation therapy Proc. SPIE Medical Imaging. 10135 374–81

MacLennan G 2017 After the plan: using templates and scripts to improve documentation Proc AAMD Annual Meeting Available from: https://pubmed.ncbi.nlm.nih.gov/28844952/

Yu V Y et al 2015 Development and verification of a highly accurate collision prediction model for automated noncoplanar plan delivery Med Phys. 42 6457–67

Brahme A, Nyman P and Skatt B 2008 4D laser camera for accurate patient positioning, collision avoidance, image fusion and adaptive approaches during diagnostic and therapeutic procedures Med Phys. 35 1670–81

Becker S J 2011 Collision indicator charts for gantry-couch position combinations for Varian linacs J Appl Clin Med Phys. 12 16–22

Becker S J, Culberson W and Flynn R T 2013 Collision indicator charts for gantry-couch position combinations for Siemens Oncor and Elekta Infinity linacs J Appl Clin Med Phys. 14 278–83

Padilla L, Pearson E A and Pelizzari C A 2015 Collision prediction software for radiotherapy treatments Med Phys. 42 6448–56

Jung H, Kurn O, Han Y, Park H C, Kim S and Choi D H 2015 A virtual simulator designed for collision prevention in proton therapy Med Phys. 42 6021–7

Cardan R A, Popple R A and Fivesh J 2017 A priori patient-specific collision avoidance in radiotherapy using consumer grade depth cameras Med Phys. 44 3430–6

Yu L, Bai J and Ni C 2018 Real-time perception of patient space for collision avoidance in radiation treatment Proc IEEE-EMBS Conf pp 629–32

Danuser S et al 2019 Surface imaging for treatment geometry virtualisation and collision detection in proton therapy PTCoG Conf Rec Available from: https://ptcog.ch/images/ptcog58/Scientific/1410_Danuser.pdf

Kikinis R, Pieper S D and Vosburgh K G 2014 3D Slicer: A Platform for Subject-Selective Image Analysis, Visualization, and Clinical Support (New York: Springer) pp 277–89

Ho M W 2019 3D visualization tool for IMPT planning PTCoG Conf Rec Available from: https://public-poster-links.s3.amazonaws.com/PTC58G/PTC58G-PTCS8-0659.pdf

GNU General Public License, GPLv3; 2007. http://gnu.org/licenses/gpl.html

3D Systems Inc. 1989 Stereo lithography Interface Specification (Valencia, CA, USA) F/ P/ N 50065–501–08

Cignoni P, Corsini M and Ranzuglia G 2008 MeshLab: an open-source 3D mesh processing system ERCIM News Available from: http://ercim-news.ercim.eu/meshlab-an-open-source-3d-mesh-processing-system

Association NEM 1993 NEMA PS3/ISO 12 052, Digital Imaging and Communications in Medicine (DICOM) Standard (VA, USA: Rosslyn) http://medical.nema.org/

Foord M and Muirhead C 2009 IronPython in Action (Greenwich, CT, USA: Manning Publishing Co.) https://dl.acm.org/citation.cfm?id=1592919

Sells C and Gheldand I 2003 Windows Forms Programming in Visual Basic. Net. (Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.) https://dl.acm.org/citation.cfm?id=940275

GitHub. MGH Radiation Oncology, 2019. https://github.com/mghro/rad-collision

Dice L R 1945 Measures of the amount of ecologic association between species Ecology 26 297–302

Shpunt A and Erlich R 2013 Scanning depth engine US20130207970A1. Available from: https://patents.google.com/patent/US20130207970A1/en

Taha A A and Hanbury A 2015 Metrics for evaluating 3D positional accuracy and clinical acceptability of 3D position and rotation information in the setup process for stereotactic radiosurgery Med Imaging 15 29

Davis A M, Pearson E A, Pan X, Pelizzari C A and Ali-Hallaq H 2019 Collision-avoiding imaging trajectories for linac mounted cone-beam CT J Xray Sci Technol. 27 1–16