gMocren: Visualization software for Monte Carlo simulators for radiotherapy

Akinori Kimura\textsuperscript{1*}, Kyoko Hasegawa\textsuperscript{2}, Ayumu Saitoh\textsuperscript{3}, Satoshi Tanaka\textsuperscript{2}

\textsuperscript{1}Faculty of Engineering, Ashikaga Institute of Technology
\textsuperscript{2}College of Information Science and Engineering, Ritsumeikan University
\textsuperscript{3}Graduate School of Engineering, University of Hyogo

\*AKimura@ashitech.ac.jp

Received: September 5, 2014; Accepted: November 14, 2014; Published: May 31, 2015

Abstract. Radiotherapy is one method of treatment for cancer. To achieve safe, accurate and efficient irradiation to the diseased portion of a patient during radiotherapy, simulation software, treatment-planning software and various treatment devices for radiotherapy have been developed. Our visualization software for radiotherapy simulations, gMocren, is described in this paper. It was developed in combination with a unique Monte Carlo simulator for radiotherapy that calculates fully physical behaviors in both the complex beam-delivery devices and the patient’s body. The visualization software creates images of the geometrical data of the beam-delivery devices, calculated physical quantities, particle trajectories, and the patient volume data set from the radiotherapy simulation. gMocren has been used to assist in the development of beam-delivery devices and simulation software. It assists users in visually and intuitively understanding the results of such simulations. The manual, an online tutorial and downloadable files for gMocren are also available to users on the gMocren website.

Keyword: visualization software, radiotherapy simulation, volume rendering

1. Introduction

Monte Carlo simulations are widely used in medical physics research [1,2]. Such simulations address phenomena that play important roles in radiotherapy to precisely evaluate dose distributions in a patient’s body or to optimize the device parameters of beam-delivery systems.

Geant4 is a C++ toolkit for the construction of Monte Carlo simulations to calculate the passage of particles - e.g., protons, electrons, and heavy ions — through matter [3]. The toolkit was originally developed for high-energy physics. Recently, the use of Geant4-based
Journal of Advanced Simulation in Science and Engineering

Simulations is increasing in medical physics, especially for radiotherapy [4,5]. However, the Geant4 developers have been forced to spend a considerable amount of time in implementing C++ code for functionalities required by radiotherapy that were not originally provided in Geant4. For instance, it has been necessary to implement the handling of input volume data, such as patient CT data. The visualization of such input data and the output of radiotherapy simulations is also important, but the original version of Geant4 does not support the visualization of volume data.

Recently, a software framework for Monte Carlo radiotherapy simulations, PTSim, has been developed [6]. PTSim was constructed based on the Geant4 toolkit. PTSim has provided a common modeling platform that is now used in three Japanese proton and ion therapy facilities and also in three additional such facilities in other countries. PTSim allows non-expert users of Geant4 to accurately and efficiently run Geant4 simulations for any of its pre-built configurations and parameter settings. PTSim includes a class library for geometric descriptions, material definitions, the optimization of physics process settings, the scoring of physical quantities, event-level parallel processing and the main program.

Figure 1: PTSim output visualized by gMocren. Incident protons from the upper right side of the image pass through the beam-delivery devices and irradiate the disease portion of the patient’s head. PTSim is a Monte Carlo simulation framework based on the Geant4 toolkit.

The original PTSim was not equipped with volume-rendering capabilities [7]. Therefore, we developed a volume visualizer, gMocren, based on the requirements for visualization
software for radiotherapy simulations as a part of the PTSim research project. Figure 1 shows PTSim output visualized by gMocren. gMocren is capable of drawing the volume data set of a patient. The beam-delivery devices, particle trajectories and calculated physical quantities, such as dose distributions, from radiotherapy simulations can also be drawn in addition to the volume data set. Users can access the manual, an online tutorial, utility tools, and downloadable files for gMocren on the gMocren website. In this paper, we describe the functionalities of gMocren and some examples of its use in detail. gMocren has the following specific features: (a) It can create a 3D fused image of a dose distribution and a human body. (b) It is capable of the fused visualization of beam-delivery devices and particle trajectories in a radiotherapy simulation in addition to 3D ray-casting volume rendering [8]. (c) Its implementation does not depend on specific hardware, meaning that it is supported on multiple operating systems, i.e., Linux, Windows and Mac OS X. (d) It is freely available with simple registration on the gMocren website. (e) gMocren is the only volume renderer for Geant4 that is used throughout most of the world for the purpose of developing radiation simulators. Because there are few such visualization software packages for radiotherapy simulations, gMocren has many users both in Japan and abroad; thus far, the total number of downloads has reached approximately 1,360.

2. Requirements for visualization software for radiotherapy simulations

The PTSim simulation framework for radiotherapy handles the passage of particles through not only a patient’s body but also the specified beam-delivery devices. The beam-delivery system controls the area to be irradiated and the energy range of the particles that are accelerated by a particle accelerator. The irradiation area is controlled to conform to the shape of a tumor in the patient’s body. A volume data set representing densities in a human body is typically obtained using a CT scanner. The known materials and shapes of each beam-delivery device in combination with such a volume data set are then used in the simulation.

PTSim calculates physical quantities based on particle interactions in the materials. It yields various physical quantities, such as the energy, momentum, trajectory, interaction points and identities of the physics processes undergone by each particle passing through the materials of the beam-delivery devices and the human body. At least 10k particle trajectories are visualized in each simulation, where a particle trajectory consists of between 100 and 10k steps. The dose distribution delivered to the tumor is calculated using those physical quantities and the geometrical information concerning the patient’s body that was obtained using the CT scanner. Visualization software for such a tool must provide the necessary functionalities and speed to process and recognize such a large quantity of data.
To draw the target region to be irradiated in the human body, it is necessary to visualize the inner portion of the volume data set. Therefore, it is necessary to introduce a volume-rendering technique [8]. In radiation simulations for clinical use, the volume data set of a human body, with voxel spacing between 2 mm and 10 mm, is used to calculate the necessary amount of irradiation to be delivered to a diseased region in a radiation treatment planning system. For instance, in the case of a volume data set of 512×512×200 voxels obtained by a CT scanner, in which the voxel size is approximately 0.5×0.5×1 mm³, the planning system calculates the amount of irradiation using volume data that has been converted in size from 26×26×20 to 128×128×100 voxels. Visualization software is, of course, necessary to draw the volume data set used by the simulation. Furthermore, it should also be possible to draw a high-resolution data set obtained using a CT scanner. The dose distribution calculated in the simulation is drawn superimposed on the CT image to allow for its evaluation. Therefore, a high-resolution image of the diseased region is required, and such an image can be obtained using the CT scanner.

Certain volume-rendering functionalities are expected to be used particularly often in the context of radiotherapy. The desired tools include a tool to generate a histogram of CT values, a tool to edit transfer functions with respect to opacities and CT values, a color-map editor, and a clipping editor to draw a cross-sectional view of a volume data set. Two-dimensional images are used for clinical and diagnostic treatment. Therefore, the ability to produce 2D images is indispensable for the drawing of sliced cross-sectional images in two dimensions. It is also desirable to be able to draw dose distribution on such a sliced image as a contour plot.

The ability to draw the beam-delivery devices is useful and important for the development of new beam-delivery devices, the assessment of adjustments to the irradiation beam, and the development of radiation simulators. It is sufficient to be able to draw the beam-delivery devices in a wireframe view or in a solid view with surface rendering. Different colors or different types of lines are generally used to represent particle trajectories that can be classified in terms of different types of particles or particles that are of particular interest in certain physics processes.

Suitable visualization software requires sufficient rendering speed that it can be operated interactively while displaying a quality image on personal computers. Moreover, it must be available on Linux, Microsoft Windows or Mac OS X because radiation simulations have been developed to run on all these operating systems.

3. Functionalities of gMocren and examples

gMocren was developed as visualization software for PTSim, which is a Geant4-based radiotherapy simulation framework. gMocren provides the requisite functionalities for radiotherapy simulation applications that are described above.
3.1. Window configuration

gMocren is a single-window application. The graphical user interface was designed using GTK+ [9]. The window consists of tool buttons, a large main pane, three cascade panes, and a pane for a histogram, color-map editor and opacity-curve editor, as shown in Fig. 2. The positions of the 3D image and the 2D images are exchangeable between the main pane and each of the cascade panes on the right. The main pane is the largest of the image panes; therefore, it can be used to enlarge any of the images. The color-map and opacity-curve editors for the patient image (upper) and the physical quantity of interest (lower) are arranged in the lower pane. These editors contain lines and histograms. The line represents the shape of the opacity curve for the voxel values, which is a transfer function. The histogram represents the accumulated voxel values.

![Figure 2: Visualization software, gMocren, for Geant4-based radiotherapy simulations. The window consists of five panes. The main pane displays a fused image of the head region of a patient drawn via volume rendering, particle trajectories, and the bolus and multi-leaf collimator of the beam-delivery devices. The three cross-sectional images in the column of panes on the right-hand side are multi-planar reformatted images with particle trajectories. The color-map and opacity-curve editors for the patient image (upper) and the physical quantity of interest (lower) are arranged in the lower pane.](image)

3.2. Opacity-curve editor and performance of 3D volume rendering

gMocren is capable of 3D volume rendering. The ray-casting engine of KGT Inc. has been
incorporated into gMocren for 3D volume rendering [10]. The ray-casting engine is not hardware dependent. Therefore, gMocren could be developed relatively straightforwardly for the Linux, Windows, and Mac OS X operating systems for personal computers. It is possible to simultaneously draw a complex patient image and the corresponding calculated dose distribution, as shown in Fig. 3. The calculated dose distribution, represented by the light blue region, is fused into the image of the chest region of the patient in 3D.

![Figure 3: A calculated dose distribution (the light blue region) fused into an image of the chest region of a patient.](image)

The rendering speed depends on the opacity curve, which represents the translucency processing in the volume rendering. Several frame rates for 3D volume rendering were measured under the two conditions represented in Fig. 4. The opacity curves for the patient and the physical quantity in Fig. 4 (a) form a slope. The opacities increase linearly with respect to the CT values. The opacity curves in Fig. 4 (b) behave as threshold functions for both the patient and the physical quantity. The left-hand side of each curve represents the transparent region, and the other side represents the non-transparent region. A case such as that in Fig. 4 (a) incurs a relatively heavy load because the number of steps processed during ray casting increases in the translucent region. The opacity curve can be manually edited using the opacity-curve editor.

In the case represented in Fig. 4 (a), the rendering speeds for the 3D images of the volume data set presented in Fig. 3 were 5.8 fps, 0.73 fps, and 0.40 fps for images of 128×128 pixels, 384×384 pixels, and 512×512 pixels, respectively. In the case represented in Fig. 4 (b), the
rendering speeds were 29 fps, 4.2 fps, and 2.2 fps, respectively for the same conditions. The frame rates were measured on a 27-inch Apple iMac (late 2013) with an Intel Core i7 3.5 GHz CPU, 32 GB of main memory, and an NVIDIA GeForce GTX 775M graphic processor. The 3D image size in gMocren is selectable from among sizes of 64×64 pixels, 128×128 pixels, 256×256 pixels, 384×384 pixels, and 512×512 pixels. For usability, the default size is 128×128 pixels for a moving image and 384×384 pixels for an image at rest.

(a) CT value

Opacity curve

(b)

Opacity

0

1

Figure 4: Two different opacity-curve settings represented by the blue lines superimposed on the histograms. (a) The opacity curves for the patient and the physical quantity form slopes. The opacities increase linearly with respect to the CT values or the values of the physical quantity, respectively. (b) The opacity curves behave as threshold functions for both the patient and the physical quantity. The left-hand side of each curve represents the transparent region, and the other side represents the non-transparent region.
3.3. 2D MPR images

The MPR (multi-planar reconstruction) in gMocren consists of three cross-sectional images in the x-y, y-z and z-x planes in the Cartesian coordinate system of a structured volume data set. The positions of the planes can be controlled using the sliders at the right-hand sides of these images. The colored, crossed lines on these images, as shown in Fig. 3, control the plane positions. On each image, the positions of these lines correspond to the positions of the other planes. When one of these lines is moved using a mouse, the plane position of the image that corresponds to that line is interactively modified. The value of the physical quantity corresponding to the location at which the mouse is pointing is displayed at the upper left corner of the image, as shown in Fig. 5.

3.4. Superposition of simulated physical quantities

Figure 5 demonstrates that gMocren is capable of superposing a physical quantity calculated in a simulation onto an MPR image. The physical quantity is drawn as a color map or a contour plot. The key for the color map or the number of contour lines can be controlled from the color-map editor or the contour-setting function, respectively.

3.5. Fused visualization of particle trajectories and beam-delivery devices with human-body images

The 3D image displayed in the main pane in Fig. 2 visualizes the head region of a patient drawn via volume rendering, the particle trajectories, and the bolus and multi-leaf collimator of the beam-delivery devices. The beam-delivery devices used in the simulation are drawn in
a wireframe view. The particle trajectories are drawn in several different colors to distinguish different types of particles or particles that are of particular interest in physics processes.

Figure 6: A cropped 3D image of head-region data. The cropping function is required for the visualization of diseased regions in the human body.

3.6. Cropping images

Figure 6 presents a cropped image of the head section of a patient data set. Such cropping is required to visualize the inner regions of a patient, such as diseased regions. The cropping position can be specified along the axes of the Cartesian coordinate system of the volume data set.

3.7. DICOM file support

gMocren partially supports the DICOM standard file format [11]. Figure 7 displays images generated from a DICOM data set. The size of the DICOM data set was 512×512×745 pixels, relatively large for a DICOM image. RoI (region of interest) data is supported not only in the GDD file format but also in the external DICOM-RT Structure Set file format, which is the standard file format. Figure 8 displays RoI images constructed from a DICOM-RT Structure Set file. These images were drawn over the MPR images of the patient. The lines were drawn using the colors specified in the file.
3.8. gMocren file format

The configuration of gMocren files is described here. A gMocren file is called a GDD file. This type of file is formatted in a dedicated configuration in binary format. As shown in Table 1, it consists of six objects: header data, modality, calculated physical quantities, RoI, particle trajectories and detectors.
Table 1: The configuration of the gMocren data format. Each data file is formatted in a dedicated configuration in binary format. It consists of six components: header data, modality, calculated physical quantities, RoI, particle trajectories and beam-delivery devices. The letters and numbers indicated in square brackets represent the data type in the C programming language and the data length in bytes, respectively. The data types are indicated as follows: char is c, unsigned char is uc, short is s, int is i, unsigned long is ul, and float is f.

| Component          | Descriptions                                                                                                                                 |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| **Header**         | File identifier [c8], file-format version [uc1], byte order [uc1], size of comment string (fixed size currently) [i4], comment string [c1024], voxel spacing of volume data set in 3D [f4x3], pointer to modality component [ul4], number of physical quantities (Nq) [i4], pointers to each physical-quantity component [ul4×Nq], pointer to RoI data set [ul4], pointer to particle-trajectory component [ul4], and pointer to beam-delivery device component [ul4] |
| **Modality**       | Dimensions of volume data set (Nx, Ny, Nz) [i4x3], min. and max. voxel values (Vmin, Vmax) [s2+s2], units of voxel values [c12], scale factor of voxel values [f4], volume data set [s2×Nx×Ny×Nz], and conversion table of H.U. values to densities [(Vmax-Vmin+1)*f4] |
| **Physical quantities** | Dimensions of volume data set (Nq1x, Nq1y, Nq1z) [i4x3], min. and max. voxel values (Vq1min, Vq1max) [s2+s2], units of volume data set [c12], scale factor of voxel values [f4], volume data set [s2×Nq1x×Nq1y×Nq1z], relative position with respect to the modality [i4×3], and name of the quantity [c80] |
| **RoI**            | Dimensions of volume data set (Nr1x, Nry, Nr1z) [i4x3], min. and max. voxel values (Vr1min, Vr1max) [s2+s2], scale factor of voxel values [f4], volume data set [s2×Nr1x×Nry×Nr1z], and relative position with respect to the modality [i4×3] |
| **Particle trajectory** | Number of particle trajectories (Npt) [i4]                                                                                               |
|                    | Number of steps (Npt1) [i4], color [uc1×3], and position of each step [f4×6×Npt1]                                                                 |
| **Beam-delivery devices** | Number of beam-delivery devices (Nd) [i4]                                                                                               |
|                    | Number of edges (Nd1) [i4], position of each edge [f4×6×Nd1], color [uc1×3], and name of the beam-delivery device [c80]                      |
|                    | … Nd iterations of the above set of contents                                                                                             |
The header contains the file information and general information: the file identifier, the file-format version, the byte order, the size of the comment string, the comment string, the voxel spacing of the volume data set in 3D, the pointer to the modality object, the number of physical quantities, the pointers to each physical-quantity object, the pointer to the RoI object, the pointer to the particle-trajectory object, and the pointer to the detector object.

The modality object consists of the dimensions of the volume data set, the minimum and maximum voxel values, the units and scale factor of the voxel values, the density or Hounsfield unit (CT) values of the volume data set, and the conversion table from Hounsfield unit values to density values.

The physical-quantities object consists of the dimensions of the volume data set, the minimum and maximum voxel values, the units and scale factor of the voxel values, the physical-quantity values of the volume data set, the relative position with respect to the modality, and the name of the quantity. Furthermore, this information is repeated for each physical quantity individually.

The RoI object contains the boundary data of the RoI regions as a volume data set. It consists of the dimensions of the volume data set, the minimum and the maximum voxel values, the scale factor of the voxel values, the boundary data of the RoI, and the relative position with respect to the modality.

The particle-trajectory object consists of the number of particle trajectories as well as the number of steps, the color, and the position of each step for each particle trajectory. A particle trajectory consists of short, concatenated steps in the Geant4-based simulation. In such a simulation, a particle may pass through various materials. The particle trajectories are separated when the particle passes on a material boundary or when a physical process occurs.

The detectors object consists of the number of beam-delivery devices as well as the number of edges, the positions of each edge, the color, and the name for each beam-delivery device. Hence, gMocren provides only wireframe rendering to draw the beam-delivery devices. Therefore, it retains sets of edge positions to represent the shape of each beam-delivery device.

4. Related software

4.1. Utility tools

The C++ class library, gMocrenIO, for gMocren file I/O is provided on the gMocren website. It is useful for allowing user-written code to create GDD files. The gMocrenIO library also includes application software: dumpgdd, makegdd, mergegdd, and dicom2gdd. Their purposes are to dump a GDD file into text format, create a GDD file from specific configured files, merge two GDD files, and convert a DICOM data set into a GDD file, respectively. The application software can also be used as sample code to demonstrate how to implement the
4.2. gMocren-file driver for Geant4

The gMocren-file driver was developed in C++ code as a graphics driver for the Geant4 visualization system. Previously, the Geant4 visualization system did not provide any driver for the visualization of volume data. Therefore, the gMocren-file driver is only the driver that enables Geant4 to handle volume data. It can be used to easily save the output of a Geant4-based simulation as a GDD file.

![Class Diagram](image)

Figure 9: The class diagram of the graphics driver for the Geant4 visualization system and the gMocren-file driver for volume visualization. The gMocren-file driver is implemented as the following derived classes: G4GMocrenFile, G4GMocrenFileSceneHandler and G4GMocrenFileViewer.

To construct the gMocren-file driver, it was necessary to implement three concrete classes derived from the G4VGraphicsSystem, G4VViewer, and G4VSceneHandler abstract base classes in the Geant4 toolkit, as shown in Fig. 9. These derived concrete classes that are depicted as unfilled arrowheads are named G4GMocrenFile, G4GMocrenFileViewer, and G4GMocrenFileSceneHandler, respectively. G4GMocrenFile is used to manage the gMocren-file driver. G4GMocrenFileViewer is used to create GDD files.
The requisite information for the creation of a GDD file is extracted from the Geant4 kernel by the G4GMocrenFileSceneHandler class and saved to the file using the G4GMocrenIO class, which is depicted as a filled diamond and a solid line. The gMocren-file driver extracts the following data:

- Modality data: a volume data set that consists of the volume dimensions, the voxel size, the density of each voxel, and the relative position with respect to the world coordinates of the Geant4-based simulation.
- Calculated physical quantities: a volume data set for each quantity that consists of the volume dimensions, the voxel size, the value of the quantity for each voxel, and the relative position with respect to the world coordinates of the Geant4-based simulation.
- Particle trajectories: each particle trajectory consists of a set of short concatenated steps, each of which is a line.
- Beam-delivery devices: the shape information for each device consists of the lines for the wireframe rendering.

![Image](https://example.com/image1.png)  
(a)  

![Image](https://example.com/image2.png)  
(b)  

Figure 10: Reconstruction results for a 3D volumetric model created from tomographic image data of the HSB. Here, (a) and (b) represent several neuropil areas and the midbrain area, respectively.

5. Application to other areas of research

gMocren has been used as visualization software not only for radiotherapy simulations but also in other areas of research. Komano et al. have used gMocren to visualize the result of their technique for
reconstructing a volume model from tomographic image data of the honeybee standard brain (HSB) [12], as shown in Fig. 10. They created a GDD file that combined multiple sets of volume data divided into neuropil areas in the HSB. Each set of volume data was assigned to its own individual modality data in the GDD file to allow the data corresponding to each neuropil area to be easily and separately visualized using the opacity-curve editor.

J. Lee et al. have used gMocren with Geant4 to compare doses calculated using Geant4 to those calculated using commercial radiotherapy treatment planning systems [13]. They note that the ability to create a 3D image via volume rendering is useful for quantitative analysis but that they require more enhanced functionalities to display contour maps for analysis.

6. User support

6.1. Website

One objective during the development of gMocren was that it be suitable for many users to use it in their research. Therefore, a gMocren website is available at http://geant4.kek.jp/gMocren/, as shown in Fig. 11. gMocren is free software. It is downloadable from the website following a simple registration procedure. The manual, an online tutorial page and utility tools are provided for users.

Figure 11: gMocren web site. Pages for the manual, an online tutorial, downloadable files and utility tools are provided. User feedback and requests can also be received through the website.
6.2. **Online tutorial**

An online tutorial page has been provided on the website to allow users to learn all of the functionalities of gMocren. There are gMocren users throughout the world. Therefore, an online tutorial enables them to interactively learn to use gMocren whenever they wish. The tutorial page was created using Adobe FLASH. The system requirements are a web browser with Adobe Flash Player 9 or later and a screen resolution of 1024×768 or higher.

The online tutorial is structured as follows:

1. Introduction
2. Installation
3. How to create GDD files
4. Initializing and reading data
5. Functionalities
6. Saving screen images and exiting

The introduction section presents the system requirements for gMocren. The installation section covers how to download and install gMocren and the installation of the GTK library, which is an external library that is necessary to run gMocren. The GTK library is used to construct the GUI of gMocren. The third section explains how to create a GDD file from a Geant4-based simulation and the usage of the utility tools provided on the website. The following functionalities of gMocren are explained in the functionalities section: the operation of the MPR panes, the opacity-curve editor, the drawing of contour plots of physical quantity, the drawing of regions of interest, the drawing of particle trajectories, the drawing of beam-delivery devices, the cropping of volume data sets, and the removal of hidden surfaces in the three-dimensional pane.

7. **Conclusions**

This paper explains and demonstrates the use of gMocren, visualization software for Geant4 simulation toolkit, especially PTSim, which is a Geant4-based radiotherapy simulation framework. gMocren is used to analyze the results of radiotherapy simulations and also for debugging Geant4/PTSim simulation software for tasks such as the refinement of beam-delivery devices as well as many other purposes. It assists users in visually and intuitively understanding the results of such simulations. gMocren has the particular feature that it is capable of simultaneously drawing devices and particle trajectories in radiotherapy simulations via 3D volume rendering. gMocren also has functionalities of drawing the calculated physical quantities, ROIs, and patient volume data sets for radiotherapy simulations. The manual, an online tutorial and downloadable files are also provided for users on the gMocren website, which has been accessed by many users throughout the world. The total number of downloads has reached approximately 1,360 since the initial release in June 2006.
Acknowledgments

The authors wish to acknowledge Takashi Sasaki, John Allison and Joseph Perl for technical assistance. This research is a part of ``The Development of Software Framework for Simulation in Radiotherapy,'' which is supported by Core Research for Evolutional Science and Technology (CREST) of the Japan Science and Technology Agency (JST). This research was supported in part by the R-GIRO research fund from Ritsumeikan University and JSPS KAKENHI Grant Number 25246044, 25280044, 26400296. We would like to thank KGT Inc. (presently Cybernet Systems Co., Ltd.) for their cooperation with the development of gMocren.

References

[1] D. J. (Jan) Van der Laan, M. C. Maas, D. R. Schaart, P. Bruyndonckx, S. Léonard, C. W. E. van Eijk: Using Cramer-Rao Theory Combined with Monte Carlo Simulations for the Optimization of Monolithic Scintillator PET Detectors, IEEE Transactions on Nuclear Science, 53:3 (2006), 1063-1070.

[2] Y. Kase, N. Kanematsu, T. Kanai, N. Matsufuji: Biological dose calculation with Monte-Carlo physics simulation for heavy-ion radiotherapy, Phys. Med. Biol., 51:24, (2006), N467-N475.

[3] S. Agostinelli et al.: GEANT4-a simulation toolkit, Nuclear Instruments and Methods in Physics Research A, 506:3 (2003), 250-303.

[4] L. Archambault et al.: Overview of Geant4 Applications in Medical Physics, in Proc. Nuclear Science Symposium Conference Record, 2003, 3, 1743-1745.

[5] T. Yamashita, T. Akagi, T. Aso, A. Kimura, T. Sasaki: Effect of inhomogeneity in a patient's body on the accuracy of the pencil beam algorithm in comparison to Monte Carlo, Physics in Medicine and Biology, 57:22 (2012), 7673-7688.

[6] T. Akagi et al.: The PTSim and TOPAS Projects, Bringing Geant4 to the Particle Therapy Clinic, Progress in Nuclear Science and Technology, 2 (2011), 912-917.

[7] J. Allison, L. Garnier, A. Kimura, J. Perl: The Geant4 Visualization System - A Multi-driver Graphics System, Int. J. Model. Simul. Sci. Comput., 4:supp01 (2013), 19.

[8] M. Levoy: Efficient ray tracing of volume data, ACM Transactions on Graphics, 9:3 (1990), 245-261.

[9] GTK+ (the GIMP Toolkit): http://www.gtk.org/, 2015.

[10] K. Mori, Y. Suenaga, J. Toriwaki: Fast volume rendering based on software optimization using multimedia instructions on PC platforms, in Proc. Computer
[11] O. S. Pianykh: *Digital Imaging and Communications in Medicine (DICOM)*, Springer, 2012.

[12] A. Komano, A. Saitoh, T. Isokawa, H. Ikeno, N. Kamimura, N. Matsui: Reconstruction of 3D Volume Model from Honeybee Standard Brain Tomographic Image Data, in *Proc. Numerical Analysis Symposium*, 2014, 38-41. (in Japanese)

[13] J. Lee, J. Kang, J. Kim, B. Kim, D. Jeong: Analysis by synthesis for estimation of dose calculation with gMocren and Geant4 in medical image, *Journal of Radiation Protection*, 37:3 (2012), 146-148.