Web-based, GPU-accelerated, Monte Carlo simulation and visualization of indirect radiation imaging detector performance

Han Dong, Diksha Sharma, and Aldo Badano[a)]
Division of Imaging, Diagnostics, and Software Reliability, Center for Devices and Radiological Health, U.S. Food and Drug Administration, Silver Spring, Maryland 20993

(Received 21 April 2014; revised 1 October 2014; accepted for publication 23 October 2014; published 18 November 2014)

Purpose: Monte Carlo simulations play a vital role in the understanding of the fundamental limitations, design, and optimization of existing and emerging medical imaging systems. Efforts in this area have resulted in the development of a wide variety of open-source software packages. One such package, hybridMANTIS, uses a novel hybrid concept to model indirect scintillator detectors by balancing the computational load using dual CPU and graphics processing unit (GPU) processors, obtaining computational efficiency with reasonable accuracy. In this work, the authors describe two open-source visualization interfaces, webMANTIS and visualMANTIS to facilitate the setup of computational experiments via hybridMANTIS.

Methods: The visualization tools visualMANTIS and webMANTIS enable the user to control simulation properties through a user interface. In the case of webMANTIS, control via a web browser allows access through mobile devices such as smartphones or tablets. webMANTIS acts as a server back-end and communicates with an NVIDIA GPU computing cluster that can support multiuser environments where users can execute different experiments in parallel.

Results: The output consists of point response and pulse-height spectrum, and optical transport statistics generated by hybridMANTIS. The users can download the output images and statistics through a zip file for future reference. In addition, webMANTIS provides a visualization window that displays a few selected optical photon path as they get transported through the detector columns and allows the user to trace the history of the optical photons.

Conclusions: The visualization tools visualMANTIS and webMANTIS provide features such as on the fly generation of pulse-height spectra and response functions for microcolumnar x-ray imagers while allowing users to save simulation parameters and results from prior experiments. The graphical interfaces simplify the simulation setup and allow the user to go directly from specifying input parameters to receiving visual feedback for the model predictions. © *2014 American Association of Physicists in Medicine.* [http://dx.doi.org/10.1118/1.4901516]

Key words: Monte Carlo simulation, GPU acceleration, indirect detector, CsI, detector models

1. INTRODUCTION

The computational modeling of medical imaging systems often requires obtaining a large number of simulated images with low statistical uncertainty that translate into prohibitive computing speeds. This is of particular relevancy to the modeling of the imaging performance in microcolumnar CsI detectors where thousands of optical photon scattering events have to be tracked per incident x-ray particle. Fortunately, the rise of hardware accelerators such as general purpose graphical processing units (GPGPUs) and the stream processing languages such as CUDA (Ref. 1) and OpenCL (Ref. 2) have enabled scientific simulations to take advantage of these architectures to accelerate performance. One such package is hybridMANTIS,[3] a Monte Carlo package for modeling indirect x-ray detectors with columnar scintillators. hybridMANTIS is an improved version of MANTIS (Ref. 4) (Monte Carlo x-rAy, electroN and opTical Imaging Simulation tool) and includes several new features such as on-the-fly column geometry and columnar crosstalk to model the columnar arrays more realistically as compared to MANTIS. Moreover, a load balancer is also implemented to dynamically allocate optical transport showers to the GPU and CPU computing cores.

While hybridMANTIS requires less memory than MANTIS and allows efficient simulation of clinical-size, large-area imaging detectors, researchers are required to setup computational experiments through an input file and to configure a GPU computing environment. Moreover, with the rise of advanced web development tools and technology, developers now have the ability to design and build rich interfaces within web browsers.[5–8] We present webMANTIS and visualMANTIS, visualization interfaces for the dynamic visualization of optical photon transport and columnar structures in large area detectors. webMANTIS is an interface that can be accessed via web pages either through a desktop or mobile device web browser and visualMANTIS is a stand-alone desktop application. The motivation behind this work is to improve the usability of hybridMANTIS in a research environment by helping a researcher setup computational experiments on a GPU computing cluster without having to know or deal with the configuration of the cluster. The visualization interfaces are dynamic as users can visualize the optical photon trajectories and columnar structures during execution. In this paper, we focus on the design and implementation of webMANTIS, visualMANTIS, and their validation.

2. METHODS

webMANTIS removes the difficulties of setting up and managing a GPU cluster for the user, the design goal is to provide a simple interface for running simulations without the need for compiling or configuring the LINUX environment. visualMANTIS is a single user implementation of webMANTIS designed as a stand-alone visualization interface. Moreover, additional visualization features such as generation of pulse-height spectra (PHS) and point-response function (PRF) images are included. In order to visualize hybridMANTIS dynamically, a modification is introduced into the original code in order to asynchronously retrieve simulation data from the GPU kernels while having minimal performance degradation effects on the original code. Both interfaces are freely available for download from https://github.com/diamfda. Readers are encouraged to view the manuals for minimum installation requirements in order to setup both interfaces. The rest of this section describes hybridMANTIS succinctly and details the work flow of webMANTIS and visualMANTIS.

hybridMANTIS uses PENELOPE (Ref. 9) and fastDETECT2 (optical transport routine)[3] in the hybrid CPU–GPU implementation. PenEasy,[10] a modular, user-friendly main program for PENELOPE including useful tally options and source models, was modified to allow direct output of location and energy deposited during x-ray and electron interactions within the scintillator. These data are then transferred by the optical transport routines in fastDETECT2. The novel GPU implementation of the physics and geometry models in fastDETECT2 provides features such as on-the-fly geometry and columnar crosstalk for modeling realistic columnar structures in large area detectors.

The physics of the optical transport consists of tracking photons until they are either absorbed in the bulk or at a surface or are detected at the sensor layer. The probability of absorption in the bulk is governed by the bulk absorption coefficient (μ_{abs}). Photons are either transmitted or reflected at columnar walls based on probabilities calculated using Snell's law and the Fresnel formulae.[4] Furthermore, a load balancer dynamically allocates optical transport showers to the CPU and GPU for computation. hybridMANTIS has shown to achieve significant performance increase up to a factor of 627 (Ref. 3) when compared to MANTIS and a speedup factor of 35 when compared to the same code running entirely in the CPU. The load balancer allows hybridMANTIS to hide hours of optical transport running time by executing in parallel with the x-ray and electron transport and shifts the computational bottleneck from optical to x-ray transport. Due to the on-the-fly geometry feature, hybridMANTIS requires less memory than MANTIS and opens the way to simulate much larger area detectors.

2.A. webMANTIS

The front-end for webMANTIS uses the three.js WebGL library,[11] along with JAVASCRIPT/jQuery, PHP, PYTHON, HTML 5, and SHELLSCRIPT and has been tested successfully on the Mozilla Firefox and Chrome web browsers. The back-end is coded in c/c++ using the libevent[12] server library, which has been utilized in a variety of distributed systems and provides efficient load balancing in a distributed environment. The usage of libevent in webMANTIS enables multiple users to setup their own computational experiments and execute them in parallel on different GPUs in a cluster. Different users can also install webMANTIS on their own GPU clusters and access the computational resources through a web browser on the desktop or mobile devices. In order to enable multiple users to utilize webMANTIS on the same cluster, GPUs are probed sequentially, starting from the most powerful one, until the first available GPU is found. In our current setup, the GPU cluster contains five NVIDIA GTX 480. webMANTIS also allows for downloading output data generated by previous hybridMANTIS executions in a zip file for future reference. The prior simulation data include optical transport statistics, PRF, and PHS output images. Currently, webMANTIS' interface only allows changing the optical transport input parameters for fastDETECT2 and not for PENELOPE.

A workflow of the process of utilizing webMANTIS is illustrated in Fig. 1. Multiple users can access webMANTIS through a web browser which takes them to the WebGL visualization. The front-end comprises of a 3D visualization of the optical photons trajectories and PHS and PRF output images generated by the simulation, moreover, a text box detailing the optical transport statistics of the simulation is also shown. By using this front-end, the user is able to modify the input arguments listed in Table I in order to setup computational experiments. The use of jQuery, PHP, and AJAX enables the user to communicate with the cluster back-end in order to support actions such as saving input arguments, simulation data, and controlling GPU execution. Once the users have setup their input arguments, webMANTIS transfers them to the back-end where the hybridMANTIS code is then executed using a single GPU

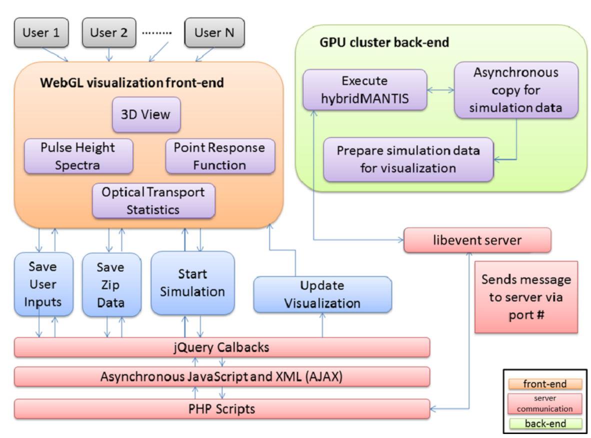

FIG. 1. High-level schematic view of the webMANTIS workflow. This describes the process in which multiple users interact with a cluster of GPUs in the back-end to execute their computational experiments. The visualizations are updated through a combination of callbacks and intermediate jQuery, AJAX, and PHP script layers in the system. visualMANTIS is similar except the back-end uses a single GPU instead of a cluster and the intermediary communications uses data files instead of scripting languages.

TABLE I. Default input parameters for simulation (parameters in bold affect visualization).

Parameters	Value
Number of x-ray histories to be simulated	100 000
Detector lateral dimensions	$909 \times 909\ \mu\text{m}^2$
Detector thickness	$150\ \mu$m
Column radius	$5.1\ \mu$m
Refractive index column	1.8
Refractive index inter-col	1.0
Top surface absorption fraction	0.1
Bulk absorption coefficient	$0.0001\ \mu\text{m}^{-1}$
Surface roughness coefficient	0.10
Minimum distance to the next column	$1.0\ \mu$m
Maximum distance to the next column	$280.0\ \mu$m
Nonideal sensor reflectivity	0.25
Number of histories to be visualized	10

and CPU in the cluster. This communication is tied together through the use of the libevent server.

Once the libevent server receives the command to execute hybridMANTIS, the input files required by hybridMANTIS are updated and code is then executed on an available GPU. webMANTIS allows one to visualize a number of optical photon trajectories on-the-fly while the simulation is still running by making a set of GPU kernel calls to simulate the optical photons. Once a kernel finishes, the simulation data are retrieved and visualized using asynchronous memory copies between GPU and CPU. For instance, as hybridMANTIS is simulating the next set of optical photons in the GPU, the retrieved simulation data are processed by the CPU before being sent back to the front-end in order to update the different visualizations. During this process, PHP scripts running in the back-end send a callback with the visualization data which travel to the WebGL visualization front-end and are shown to the user.

2.B. visualMANTIS

visualMANTIS is coded in c/c++ using the OpenGL and Fast Light ToolKit[13] libraries and is a stand-alone application that requires compilation and execution in a LINUX-based environment. During execution, files containing optical transport statistics, PRF, and PHS output images are generated. Similar to webMANTIS, the interface only allows making changes to the optical transport input parameters of fastDETECT2. The workflow diagram of visualMANTIS is similar in implementation to Fig. 1, as the back-end uses a single GPU instead of a cluster and the intermediary communications use data files instead of scripting languages. The front-end of visualMANTIS is similar to webMANTIS in displaying the generated images and photon trajectory visualizations.

2.C. Visualization parameters

Table I lists input parameters that hybridMANTIS requires. The parameters highlighted in bold represent the relevant parameters that a user can change in order to affect the visualization of the data. The other parameters are also modifiable, although they do not have a direct impact on the visualization. The number of x-ray histories for simulation specify the total number of primaries to be simulated. The detector dimensions represent the size of the scintillator. The columnar radius specifies the radius of each column in microns. It should be noted that the detector dimensions and columnar radius dimensions are not shown to scale in the visualization. Top surface absorption fraction of 0.1 indicates a reflective surface with only 10% absorption. The surface roughness coefficient indicates the degree of roughness of the column walls, with zero representing perfectly smooth walls. The distance between subsequent columns visited by the optical photon is sampled uniformly between the specified minimum and maximum distances. The number of histories to be visualized represents the number of optical photon histories for which every interaction is saved in order to view the trajectories. This parameter has a default value of ten, i.e., the first ten optical photons simulated by hybridMANTIS are retrieved using the asynchronous memory copy method described in Sec. 2.B. Users can modify this variable to visualize more histories, the only caveat being the amount of memory required to save the data. Each photon in the x-ray history is represented with five single precision floating point numbers. However, given the nondeterministic nature of the simulation, the memory required for each x-ray can range from 200 bytes to 4 kilobytes.

2.D. Validation

Although both webMANTIS and visualMANTIS are interfaces for setting up a hybridMANTIS simulation, the original hybridMANTIS GPU code was modified in order to support asynchronous data copies. We validated our results from webMANTIS against hybridMANTIS for a 40 kVp input spectra. The material geometry contains the following materials: 1 mm graphite substrate, 2 μm aluminum polish, 174 μm cesium iodide, 10 μm photodiode layer, and a 4.5 cm thick glass slab. For the purpose of validation, we used the same random input seed for both hybridMANTIS and webMANTIS, and since webMANTIS is built on top of hybridMANTIS without changing any simulation properties, thus we achieved exactly the same results for both. We calculated the Swank factor for webMANTIS to be 0.9, which matches with that of hybridMANTIS published data.[14] Thus, we conclude that webMANTIS does not change the physics of hybridMANTIS and extends its usability through a visualization.

3. RESULTS

In this section, we describe and illustrate the user interface of webMANTIS along with the corresponding visualization and sample use case of how modifying the input arguments affect the results. Since the only difference between the web and visual versions of the tool is the graphical programming language, we chose for this note to provide details and illustrate only the webMANTIS tool.

We also demonstrate validation data by comparing to published results.[14] The material geometry used in Sec. 3.A is

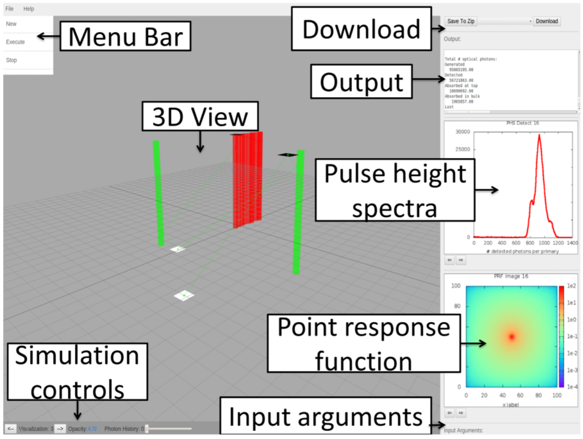

FIG. 2. Illustration of the interface of webMANTIS and labeled sections. Cylinders indicate various columns in the scintillator, while the squares mark the top and bottom surfaces where the optical photons can get reflected back or absorbed.

a 0.015 cm cesium iodide scintillator screen. We show results for x-ray scintillator detectors based on cesium iodide. Modeling other materials is possible with these tools but requires simple changes to the scripts and files in the server side. The reader interested in such changes should follow the instruction in the hybridMANTIS documentation.

3.A. webMANTIS visualizations

Figure 2 illustrates the resulting visualization with webMANTIS using the default input parameters and highlights the different working areas in the graphical user interface. webMANTIS provides a main 3D visualization in the middle of the interface. We render the history of each optical photon as representative of the lines connecting colored dots along with cylinders. A line between two dots in Fig. 2 shows the path that the optical photon has taken. The colored cylinders represent the columns in the scintillator; the optical photons can either get absorbed or reflected at the top and bottom surfaces of the scintillator marked by black and white squares. The PHS and PRF images are displayed on the right of the interface as highlighted. A button is provided that allow users to scroll through them and view images generated during the simulation. The text box situated on top right provides the optical photon transport statistics and is updated dynamically as hybridMANTIS is executed in the GPU cluster, downloading the results as a zip file. The menu bar situated at the top of the interface allows the user to control different aspects of executing the code such as starting a new hybridMANTIS job on the GPU cluster. Moreover, since this is a multiuser environment, the menu bar enables the user to manually delete their current job at any time in order avoid hogging up system resources. The download button provided above the simulation statistics text box enables the user to download prior simulation data including the statistics, PRF, and PHS images and their corresponding data files. In the bottom of the interface, webMANTIS has sliding bars in order to follow the optical photon trajectories in a step-by-step fashion. In addition, the user can alter the transparency of the graphical objects and browse through different optical photon histories.

4. CONCLUSION

We describe the visualization interfaces webMANTIS and visualMANTIS for hybridMANTIS, a Monte Carlo software package for x-ray, electron, and optical photon transport. webMANTIS abstracts away the difficulties of setting up and managing a GPU cluster for the use and provides simple interface for running computational experiments in a multiple user environment through a web browser. Although users can interact with the visualizations while the GPU is executing in the background, the download and zip feature needs to be performed manually at the end of simulation. visualMANTIS is designed and built as a stand-alone visualization application that provides a intuitive graphical user interface to setup computational experiments on a personal workstation. Both tools provide visualization features such as on-the-fly generation of PHS and PRF images, allowing users to save simulation parameters and results from prior experiments. The graphical interfaces also simplify the simulation setup and allow the user to go directly from specifying input parameters to receiving visual feedback. The utility of these tools lies primarily in the early design stages of x-ray imaging detectors during which the thickness of the CsI microcolumnar layer and the reflectivity of key surfaces can be modified by the manufacturer to obtain a desired imaging performance to match the needs of the application.

[a]Author to whom correspondence should be addressed. Electronic mail: aldo.badano@fda.hhs.gov

[1]J. Nickolls, I. Buck, M. Garland, and K. Skadron, "Scalable parallel programming with CUDA," Queue **6**(2), 40–53 (2008).
[2]J. Stone, D. Gohara, and G. Shi, "OpenCL: A parallel programming standard for heterogeneous computing systems," Comput. Sci. Eng. **12**, 66–73 (2010).
[3]D. Sharma, A. Badal, and A. Badano, "hybridMANTIS: A CPU–GPU Monte Carlo method for modeling indirect x-ray detectors with columnar scintillators," Phys. Med. Biol. **57**(8), 2357–2372 (2012).
[4]A. Badano and J. Sempau, "MANTIS: Combined x-ray, electron and optical Monte Carlo simulations of indirect radiation imaging systems," Phys. Med. Biol. **51**(6), 1545–1561 (2006).
[5]Y. Sun, B. Shen, Z. Lu, Z. Jin, and X. Chi, "GridMol: A grid application for molecular modeling and visualization," J. Comput.-Aided Mol. Des. **22**(2), 119–129 (2008).
[6]M. Latendresse and P. D. Karp, "Web-based metabolic network visualization with a zooming user interface," BMC Bioinf. **12**(1), 176–185 (2011).
[7]F. Vali and R. Hong, "A web based radiation oncology dose manager with a rich user interface developed using AJAX, ruby, dynamic XHTML and the new Yahoo/EXT user interface library," in *Annual Symposium Proceedings/AMIA Symposium, AMIA Symposium* (2007), p. 1144.
[8]P. Bourgeat, V. Dore, V. Villemagne, C. Rowe, O. Salvado, and J. Fripp, "MilxXplore: A web-based system to explore large imaging datasets," J. Am. Med. Inf. Assoc. **20**(6), 1046–1052 (2013).
[9]F. Salvat, J. M. Fernández-Varea, and J. Sempau, PENELOPE-*2008: A Code System for Monte Carlo Simulation of Electron and Photon Transport* (OECD Nuclear Energy Agency, Issy-les-Moulineaux, France, 2009) (available in pdf format at http://www.nea.fr).
[10]J. Sempau, A. Badal, and L. Brualla, "A PENELOPE-based system for the automated Monte Carlo simulation of clinacs and voxelized geometries application to far-from-axis fields," Med. Phys. **38**(11), 5887–5895 (2011).
[11]T. Parisi, *WebGL: Up and Running* (O'Reilly Media, 2012).
[12]N. Provos, "libevent—An event notification library," 2003 (available URL: http://libevent.org).
[13]B. Spitzak *et al.*, "Fast light toolkit," Bill Spitzak and others, c1998–2008 [cit. 2009-05-10], 2011 (available URL: http://www.fltk.org/documentation.php).
[14]D. Sharma and A. Badano, "Comparison of experimental, MANTIS, and hybridMANTIS x-ray response for a breast imaging CsI detector," in *Breast Imaging* (Springer, Berlin, Heidelberg, 2012), pp. 56–63.