hybridMANTIS v1.0

Diksha Sharma, Andreu Badal and Aldo Badano

Center for Devices and Radiological Health,
U.S. Food and Drug Administration,

diksha.sharma@fda.hhs.gov

Last updated on: April 19, 2012

1 Introduction

hybridMANTIS is a Monte Carlo tool for modeling x-ray detectors with columnar scintillators. It uses a novel hybrid approach to maximize the utilization of computing resources like CPUs and GPUs in modern workstations.

hybridMANTIS was developed using the CUDA programming model from NVIDIA [1] to achieve maximum performance on NVIDIA GPUs. The code can also be compiled with a standard C compiler to be executed in a regular CPU.

The main developer of hybridMANTIS is **Diksha Sharma**, at the **U.S. Food and Drug Administration**. The source code is free and open software in the public domain, as explained in the Disclaimer section below. The software distribution website is: `http://code.google.com/p/hybridmantis/`. hybridMANTIS has been described in a few publications [2,3]. The main reference for this work which should be used for citations is [2]:

```
   D. Sharma, A. Badal and A. Badano, hybridMANTIS: a CPU-GPU Monte
Carlo method for modeling indirect x-ray detectors with columnar
scintillators, Physics in Medicine and Biology, 57(8), p. 2357-72 (2012)
```

This code is still in development, please report to the author any issue/bug that you may encounter. Feel free to suggest improvements to the code too.

2 Disclaimer

This software and documentation (the "Software") were developed at the Food and Drug Administration (FDA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17, Section 105 of the United States Code, this work is not subject to copyright protection and is in the public domain. Permission is hereby granted, free of charge, to any person obtaining a copy of the Software, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, or sell copies of the Software or derivatives, and to permit persons to whom the Software is furnished to do so. FDA assumes no responsibility whatsoever for use by other parties of the Software, its source code, documentation or compiled executables, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. Further, use of this code in no way implies endorsement by the FDA or confers any advantage in regulatory decisions. Although this software can be redistributed and/or modified freely, we ask that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

3 Code features

hybridMANTIS uses PENELOPE 2006 [4] for the x-ray and electron transport with the penEasy [5] main program and tallies, and fastDETECT2 for the optical transport. fastDETECT2 is a new and improved version of the DETECT2 (optical transport used in MANTIS [6]). It offers new features like *on-the-fly geometry* and *columnar crosstalk* to model the columnar arrays in the detector more realistically.

The penEasy program [5] was modified to output the energy and location of individual energy deposition events which are then used as input to fastDETECT2 which then spawns independent optical transport kernel calls.

Load balancing: A load balancer is implemented which dynamically allocates optical transport showers to one GPU card and one CPU core. The algorithm is implemented in the `hybridMANTIS_tallyEnergyDepositionEvents.f` file. Load balancing is executed only when hybridMANTIS is run on the GPU along with the CPU. The load balancer initially runs optical transport on the GPU and CPU sequentially to obtain the corresponding GPU and CPU speeds. The PENELOPE speed is hard coded in the `hybridMANTIS_tallyEnergyDepositionEvents.f` file. Using these three speeds, the load balancing factor is computed which determines the optimal number of energy deposition events events to be sent to the GPU and CPU such that the GPU finishes processing the buffer at the moment PENELOPE in the CPU is ready to send the new load (buffer).

The hard coded PENELOPE speed (794 histories/sec) was obtained by running only the x-ray/electron transport on one core of an Intel® Core i7 920 CPU. This speed may vary if the hybridMANTIS is run on a different CPU, in which case the user needs to modify the '`bufspeed`' variable on line 208 in file `hybridMANTIS_tallyEnergyDepositionEvents.f` and re-compile.

hybridMANTIS uses PENELOPE RANECU for generating pseudo random number sequences. In order to ensure that the simulated photon tracks are not correlated, each thread initializes the generator to a unique position in the random sequence, far enough from the other threads using the algorithm described in the seedMLCG code [7].

Any x-ray/electron transport package can be used with fastDETECT2 as long as the energy deposition events information is buffered and sent to fastDETECT2 in the correct format (x,y,z,E). Also the load balancing implementation from hybridMANTIS_tallyEnergyDepositionEvents.f file will have to be copied in the new package's code.

4 Code input

hybridMANTIS requires two input files: one for running penEasy and other for running fastDETECT2. penEasy_CsI_input.in is the input file for running x-ray/electron transport, and hybridMANTIS_input.in is the input file for running the optical transport.

hybridMANTIS_input.in needs 24 input arguments for running fastDETECT2. This input file name is fixed and should not be changed. These arguments specify the simulation parameters like number of histories to be simulated, optical transport and output image parameters - point response function (PRF) and pulse height spectrum (PHS). The arguments include:

- Number of x-ray histories to be simulated
- Pulse Height Spectrum: min, max # optical photons detected, # of bins for the histogram
- Detector x, y, z dimensions (in microns)
- Detector column radius (in microns)
- Refractive index of columns
- Refractive index of inter-columnar space
- Top surface absorption fraction (range 0 to 1)
- Bulk absorption coefficient (in 1/microns)
- Surface roughness coefficient (range 0 to 1)
- Minimum and maximum sampling distance to the next column (in microns)
- PRF image: x, y lower and upper bounds (in microns), pixel pitch
- Light yield (/eV)
- Non-ideal sensor reflectivity (range 0 to 1)
- Machine Flag: run on GPU (1) or CPU (0)
- Machine number

The number of histories to be simulated is read as an input from both the input files (penEasy_CsI_input.in, hybridMANTIS_input.in). This number should be same in both the files, else the number of histories to be simulated is taken from the penEasy_CsI_input.in file.

The maximum number of bins in the PHS can be 1000, and the PRF image can have maximum 501 × 501 pixels. The number of pixels can be calculated by (upper bound - lower bound)/pixel pitch.

The bounds of PRF image are useful when one wants to generate output image for only a certain section of the sensor plane. The machine number is useful when running simulations with same input parameters on different computing resources, to ensure that the output file names are different. Most of these values are appended in the output files separated by an underscore.

Load balancing is executed only when machine flag = 1, i.e. run on GPU in parallel with CPU. When machine flag = 0, everything will run only on CPU.

5 Code output

At the end of the simulation, the code outputs the PRF image and PHS, along with the detailed optical transport statistics like number of optical photons generated, detected at the sensor plane, absorbed at the top surface of detector or in the bulk, lost at detector boundaries, the load balancing factors, total simulation time and speed.

PRF image file name starts with '`myimage_`' and PHS with '`detected_`'. Some simulation parameters are then appended at the end of the file name separated by an underscore. GNUPLOT scripts are attached to make a 2D plot of the PRF image and plot the PHS. Modify these scripts to enter the output PRF and PHS file names.

6 Code compilation and execution

hybridMANTIS has been tested only on the Linux operating system. The CUDA libraries, GNU gcc and gfortran compiler and GNU scientific library needs to be pre installed before running hybridMANTIS. A bash script is included to compile the CUDA and C codes with PENELOPE and penEasy files. This file may have to be edited to modify the library paths. A pre-compiled executable is also attached. It was compiled using CUDA version 4.0, gcc version 4.4.5 and gsl version 1.14.

To run hybridMANTIS:

```
hybridMANTIS_ver1_0.x < penEasy_CsI_input.in
```

Sample inputs and outputs for simulating 100,000 x-ray histories are included under the Example folder. PENELOPE 2006 source code files `penelope.f` and `pengeom.f` are not distributed with hybridMANTIS package, but are needed for compiling. If these files are needed, please contact Diksha Sharma at Diksha.Sharma (at) fda.hhs.gov.

References

1. NVIDIA Corporation. NVIDIA CUDA(TM) Programming Guide,. Technical Report available at www.nvidia.com/cuda, 2011.
2. D. Sharma, A. Badal, and A. Badano. hybridMANTIS: a CPU-GPU Monte Carlo method for modeling indirect x-ray detectors with columnar scintillators. *Physics in Medicine and Biology*, 57(8):2357–72, 2012.
3. D. Sharma, A. Badal, Y. Fang, and A. Badano. hybridMANTIS: a novel method for faster Monte Carlo simulation of x-ray imaging detectors. *Medical Physics*, 38(6):3834, 2011.
4. F. Salvat, J. Fernández-Varea, and J. Sempau. PENELOPE-2006: A code system for Monte Carlo simulation of electron and photon transport. *NEA-OECD*, available at http://www.oecd-nea.org/science/pubs/2006/nea6222-penelope.pdf, 2006.
5. 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. *Medical Physics*, 38:5887–95, 2011.

6. A. Badano and J. Sempau. MANTIS: combined x-ray, electron and optical Monte Carlo simulations of indirect radiation imaging systems. *Physics in Medicine and Biology*, 51(6):1545–61, 2006.
7. A. Badal and J. Sempau. A package of Linux scripts for the parallelization of Monte Carlo simulations. *Computer Physics Communications*, 175(6):440–50, 2006.