VisualMANTIS Manual Version 1.0

Updated: 7/9/2013

by

Han Dong, Diksha Sharma and Aldo Badano

Aldo.Badano@fda.hhs.gov

Center for Devices and Radiological Health,

U.S. Food and Drug Administration

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

This manual contains information on compiling and executing *visualMANTIS* visualization application. The application is written in C and the library dependencies required to build and execute are elaborated in the sections below. *visualMANTIS* is built on top of *hybridMANTIS* (Sharma, Badal, & Badano 2012); it is an extension by providing a visual way of interacting and viewing the simulation.

The chapters below will instruct the user on the dependencies required to build the application; along with steps for execution and usage. The last chapter provides information on the various aspects of the visualization and how the user can interact with them.

Chapter 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 what- soever 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.

Chapter 3

PREREQUISITES

List of dependencies:

- Some flavor of Linux e.g. *www.ubuntu.com*
- **hybridMANTIS** @ *http://code.google.com/p/hybridmantis/*
- **gfortran**
- **CUDA** 4.2
- GNU Scientific Library (**GSL**)
- Fast Light Tool Kit (**FLTK**) @ *www.fltk.org*
- **gnuplot**

3.1 Linux Instructions

At the moment, the application is only capable of being built and executed on Linux platforms. Any flavor of Linux is fine as long as you can compile and use the different libraries **hybridMANTIS** and **visualMANTIS** rely on. This application was built and tested on **Ubuntu** 11.04 64 Bit.

After downloading and installing **Ubuntu**, open a shell and execute the following commands to ensure the basic libraries are updated and installed:

- sudo apt-get update && sudo apt-get upgrade
- sudo apt-get install g++
- sudo apt-get install gdb
- sudo apt-get install subversion
- sudo apt-get install autoconf
- sudo apt-get install libx11-dev
- sudo apt-get install libglu1-mesa-dev
- sudo apt-get install build-essential gcc-4.4 g++-4.4 libxi-dev libxmu-dev freeglut3-dev

3.2 hybridMANTIS Instructions

visualMANTIS relies on the existing libraries and code that exists in **hybridMANTIS**, so it is important to have a working implementation of **hybridMANTIS** ready. In order to compile and execute **hybridMANTIS**, the libraries in the following subsections are required.

3.2.1 Installing gfortran

On Ubuntu Linux, gfortran can be installed with *sudo apt-get install gfortran*.

3.2.2 Nvidia CUDA Instructions

There are a wide range of documentation and guides online for installing CUDA on various Ubuntu versions and it is highly suggested the user go through them. It is imperative that the CUDA SDK is installed as hybridMANTIS relies on libraries in the SDK to compile correctly.

3.2.3 Installing GNU Scientific Library (GSL)

On Ubuntu Linux, libgsl can be installed with *sudo apt-get install libgsl0-dev*

3.2.4 Fast Light ToolKit (FLTK)

Fast Light Tool Kit is a simple and efficient GUI builder in C and can be downloaded from *www.fltk.org*. Follow the instructions in the README for FLTK to build and install on Linux platforms; the visualization is using FLTK 1.3.1.

3.2.5 gnuplot

Gnuplot is required for plotting purposes and can be downloaded using *sudo apt-get install gnuplot*. The visualization was successfully tested with gnuplot 4.6.

Chapter 4

BUILDING

After all of these libraries have been installed and verified to be working. It is possible to proceed to compile and execute hybridMANTIS. In the hybridMANTIS folder, there should be a *compile_ver1_0.sh* file, this file contains the script to compile and link the entire program together. Before it can be executed, it must be modified in order to link different libraries. Some libraries that need to changed are:

- -I/usr/local/cuda/include
- -I/home/user/NVIDIA_GPU_Computing_SDK/C/common/inc
- -L/usr/local/cuda/lib64/
- -L/home/user/NVIDIA_GPU_Computing_SDK/C/lib

The libraries above are dynamically linked during compilation and the locations are dependent on the where CUDA and the CUDA SDK are installed.

The following steps are required to merge the **visualMANTIS** and **hybridMANTIS** code together.

- Ensure that all libraries in Chapter 3 are working and built correctly.
- Copy all files and folders from visualMANTIS_v1.0 to the **hybridMANTIS** folder.

- Compile with *sh compile_visualmantis_ver1_0.sh*

Chapter 5

VISUALIZATION

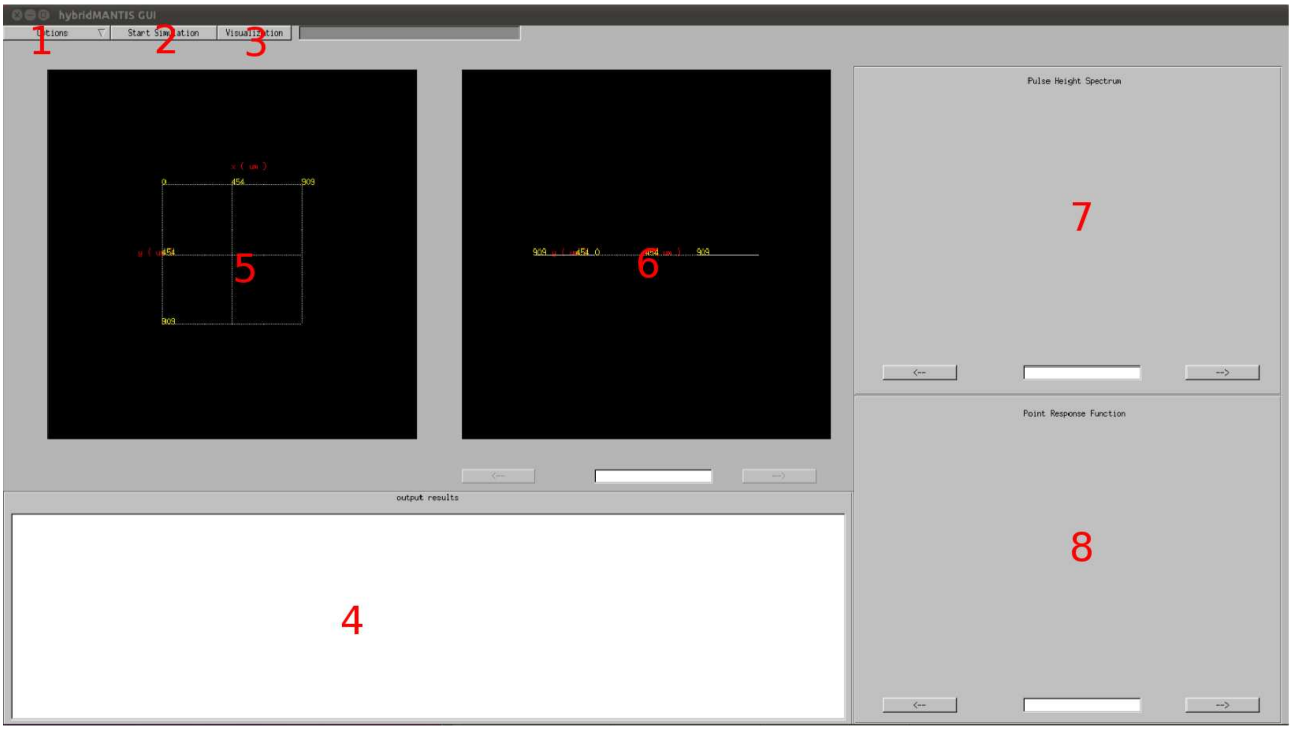

FIG. 5.1. Shows the various aspects of the visualization. There are 8 different portions that make up **visualMANTIS** and they will be explained separately below. The 8 portions are: **Options**, **Start Visualization**, **Mini-Visualization**, **Output Results**, **2-Dimensional Top Down View**, **3-Dimensional View**, **Pulse Height Spectrum**, **Point Response Function**

5.1 Options Button

The **Options** button allows the user to change different simulation parameters. The parameters here are exactly the same as the parameters described in **hybridMANTIS**; more information is listed in the **hybridMANTIS** manual. Save enables the parameters to be saved prior to executing and **Cancel** discards the changes. **Hide/Show More** displays additional input parameters that are shown in Figure 5.6. The **Number Photon Histories Visualization** field indicates the number of photon histories to save out of the initial 100000 histories specified by **X-Ray Histories**.

The following list contains the input parameters name along with a detailed description.

- **X-Ray Histories** : number of x-ray histories to be simulated (N)
- **Min Detect** : min. number of optical photons that can be detected
- **Max Detect** : max. (N*yield) number of optical photons that can be detected
- **Number of Bins** : number of bins for storing pulse height specturm (maximum value=1000)
- **X-Dimension** : x-dimension of detector (in microns)
- **Y-Dimension** : y-dimension of detector (in microns)
- **Detector Thickness** : thickness of detector (in microns)
- **Column Radius** : column radius (in microns)
- **Column Refractive Index** : refractive index of column material
- **Inter-Columnar Refractive Index** : refractive index of inter-columnar material
- **Top Surface Absorption Fraction** : top surface absorption fraction
- **Bulk Absorption Coefficient** : bulk absorption coefficient (in 1/microns)

- **Surface Roughness Coefficient** : surface roughness coefficient
- **Minimum Distance Next Column** : minimum distance to the next column (in microns)
- **Maximum Distance Next Column** : maximum distance to the next column (in microns)
- **PRF Image X Lower Bound** : x lower bound of PRF image
- **PRF Image Y Lower Bound** : y lower bound of PRF image
- **PRF Image X Upper Bound** : x upper bound of PRF image
- **PRF Image Y Upper Bound** : y upper bound of PRF image
- **Light Yield** : light yield (/eV)
- **Pixerl Pitch** : pixel pitch (in microns) (max. pixels allowed in PRF image are 501x501. calculate this by upper bound - lower bound/pixel pitch.)
- **Non-Ideal Sensor Reflectivity** : non-ideal sensor reflectivity
- **GPU (1) or CPU (0) flag** : flag for running in the GPU (1) or only in the CPU (0)
- **Machine Number** : this is helpful when running several simulations with exactly same optical parameters; the output file names contain the optical parameters with machine number appended at the end of file name.
- **Number Photon Histories Visualization** : number of photon histories to save for visualization (e.g. 10)

5.2 Start Visualization

After making changes in the **Options** window above, clicking **Start Simulation** will start executing **hybridMANTIS** and show the visualizations in the windows below.

5.3 Mini-Visualization Button

The **Mini-Visual Window** is designed to always display the first 10 photon histories. This is so that the user would not need to wait for the simulation to complete before interacting with some initial photon histories if a high number of photon histories are selected for visualization.

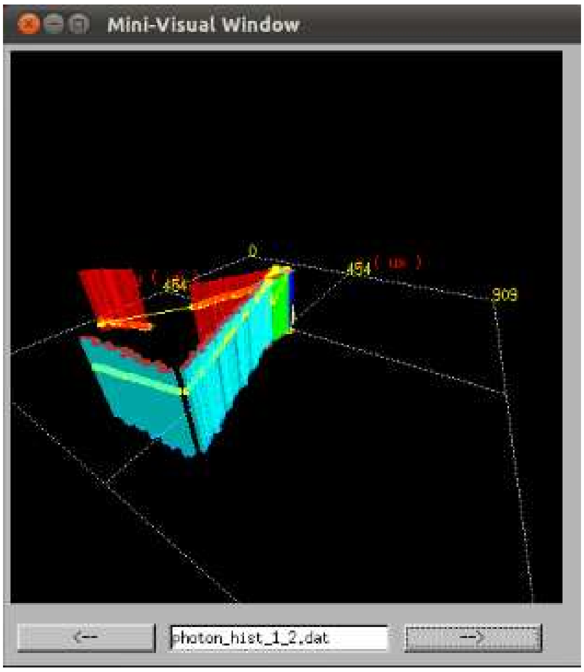

FIG. 5.2. Mini-Visual Window show some photon histories.

5.4 Output Results

Figure 5.3 shows the direct output of **hybridMANTIS** during execution and is useful as a guide-line in the simulation time-line.

FIG. 5.3. Output showing the results at end of simulation run.

5.5 2-Dimensional Top Down View

The 2D window shows the cylinders that the photons hit from a top down view. It is window number 5 in Figure 5.1.

5.6 3-Dimensional View

The 3D window is similar to the Mini-Visual Window except that it keeps a history of all photon histories. It allows the user to interact with it through the use of a mouse in rotating and zooming in regions of interest. The different colored cylinders indicate that different photons passed through it. As a photon can potentially hit the ceiling and floor and perish, it is plausible for the CUDA kernels to generate a new photon that start another process of interacting with the materials. These different colored cylinders represent that.

5.7 Pulse Height Spectrum

Figure 5.4 shows the **Pulse Height Spectrum** window, it represents images generated by **GNU-PLOT** through the detection dataset from **hybridMANTIS**.

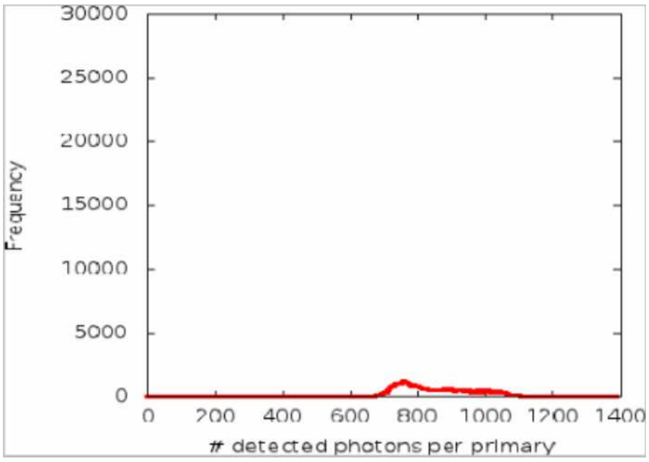

FIG. 5.4. Image generated from detection data set.

5.8 Point Response Function

Figure 5.5 shows the **Point Response Function**, it represents images generated by **GNUPLOT** and uses the imaging dataset produced by **hybridMANTIS**.

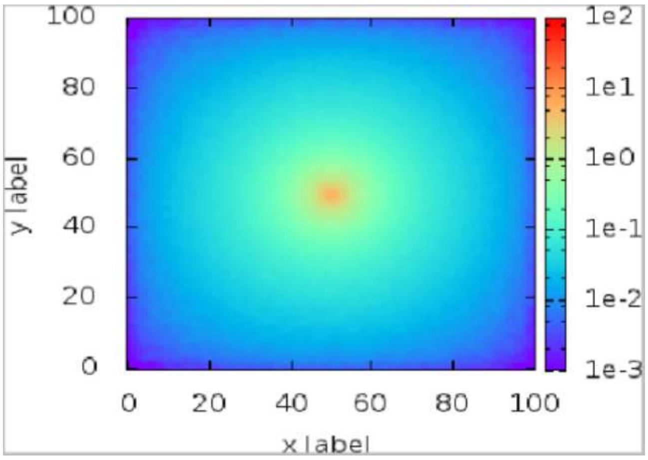

FIG. 5.5. Image generated from imaging data set.

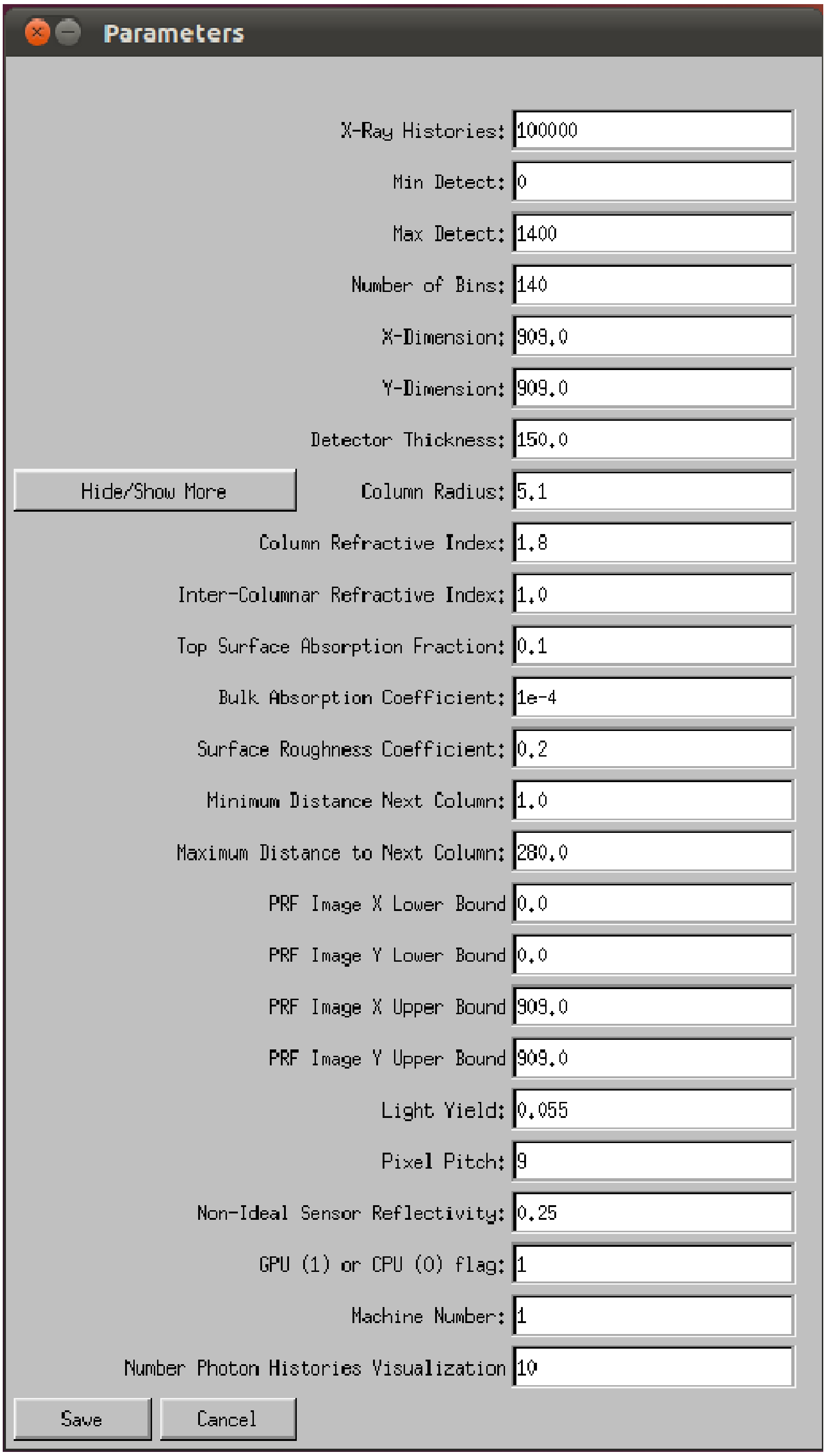

FIG. 5.6. Options menu for parameter input.

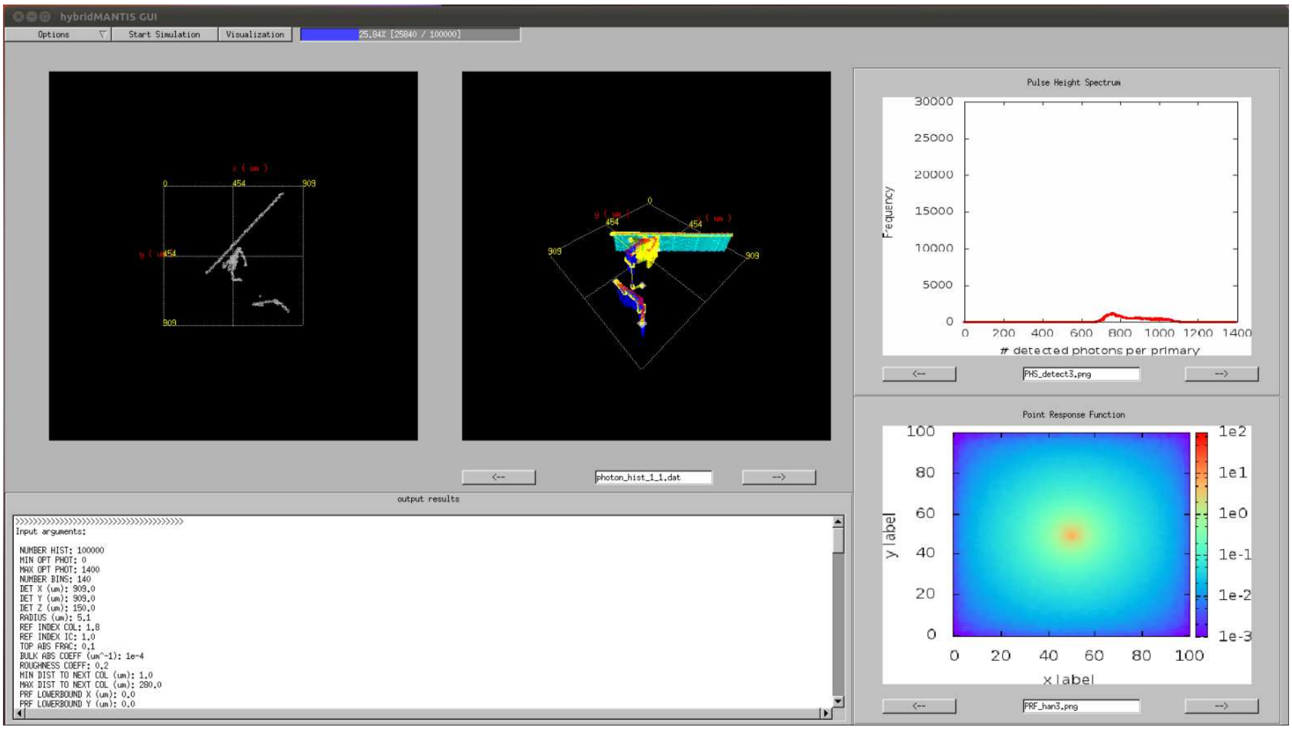

FIG. 5.7. **visualMANTIS** during execution

REFERENCES

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