

cartesianDETECT2

Diksha Sharma and Aldo Badano

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

diksha.sharma@fda.hhs.gov

Last updated on: Oct 30, 2013

1. Introduction

cartesianDETECT2 is a dedicated Monte Carlo optical transport code for modeling pixelated scintillator structures. It is based on DETECT2 (the optical transport code for MANTIS) but has been specifically tailored for use with pixelated detector structures.

The main developer of cartesianDETECT2 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 reference to be used for citing this work is

D. Sharma, C. Sze, H. Bhandari, V. Nagarakar and A. Badano. Depth-of-interaction estimates in pixelated scintillator sensors using Monte Carlo techniques, submitted to Medical Physics, 2013.

This code is still under development; please report to the author any issue or bug that you may encounter. Feel free to suggest any improvements in 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 input and output

`cartesianDETECT2` requires 18 command line arguments which specify the optical transport simulation parameters. They are (in order):

- X, Y detector dimensions (μm)
- Side and thickness of column (μm)
- Intercolumnar gap (μm)
- Refractive index of column and intercolumnar space
- Top surface absorption
- Bulk absorption (μm^{-1})
- Surface roughness coefficient
- Lower x, y PRF (Point Response Function) bounds (μm) for the image
- Upper x, y PRF bounds (μm)
- Pixel pitch (μm)
- Seed input for RNG
- Scintillation events file name (input)
- PRF file name (output)

- Radial response file name (output)

Apart from this, `cartesianDETECT2` also uses one input file to supply the information required to generate optical photons, and produces two output files.

Input file contains the energy deposition event information in terms of location in the detector (x , y , z) in μm and energy deposited (E) in eV. The number of optical photons are sampled using a Poisson distribution with a mean equal to the ratio of the energy deposited at each interaction location and the mean energy required to generate an optical photon in the CsI detector. Any available package can be used for doing the x-ray and electron transport, as long as it can give these energy deposition events. For the examples included, we used `PENELOPE 2006` with some modifications to `penEasy` main program to return the required energy deposition information per interaction event.

Output files: PRF (point response function) data and radial response on the PRF data.

Radial response on the PRF is calculated by aggregating the detected optical photons at the sensor plane in radial bins of $10\ \mu\text{m}$. To account for the change in the bin area moving outward, each bin is normalized by its area.