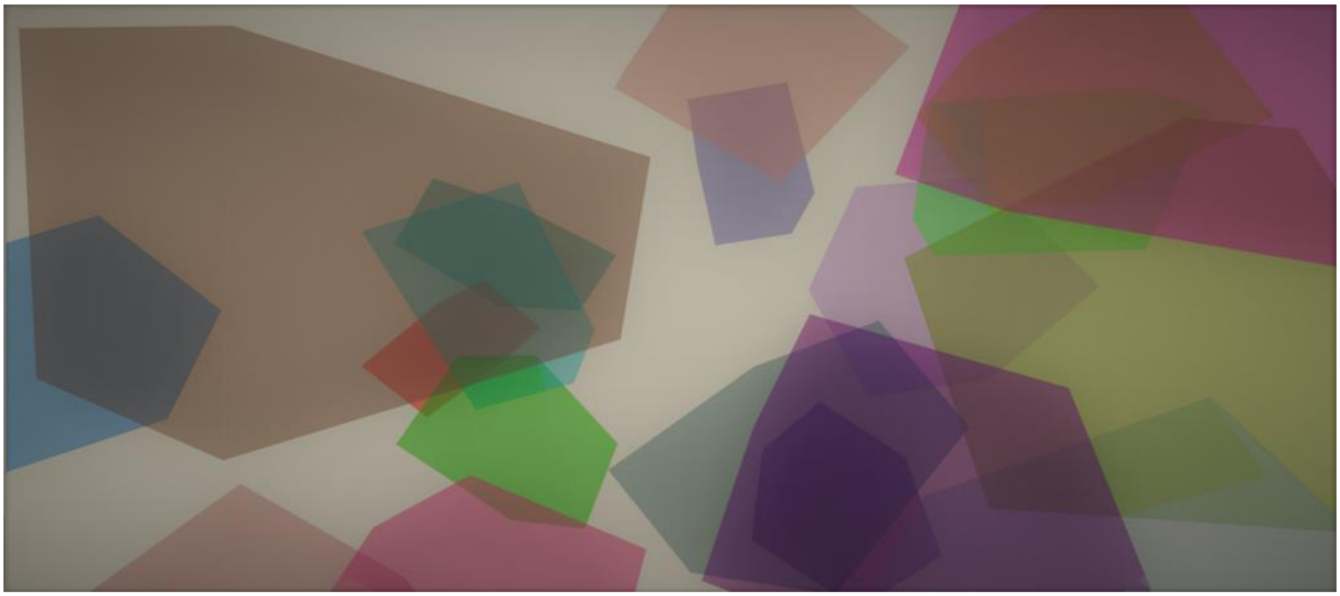


Bio-Phantom Explorer

Eurorisk Systems Ltd.
31, General Kiselov str.
BG-9002 Varna, Bulgaria
Phone +359 52 612 367
Mobile +359 52 612 371
info@euroriskssystems.com
www.euroriskssystems.com

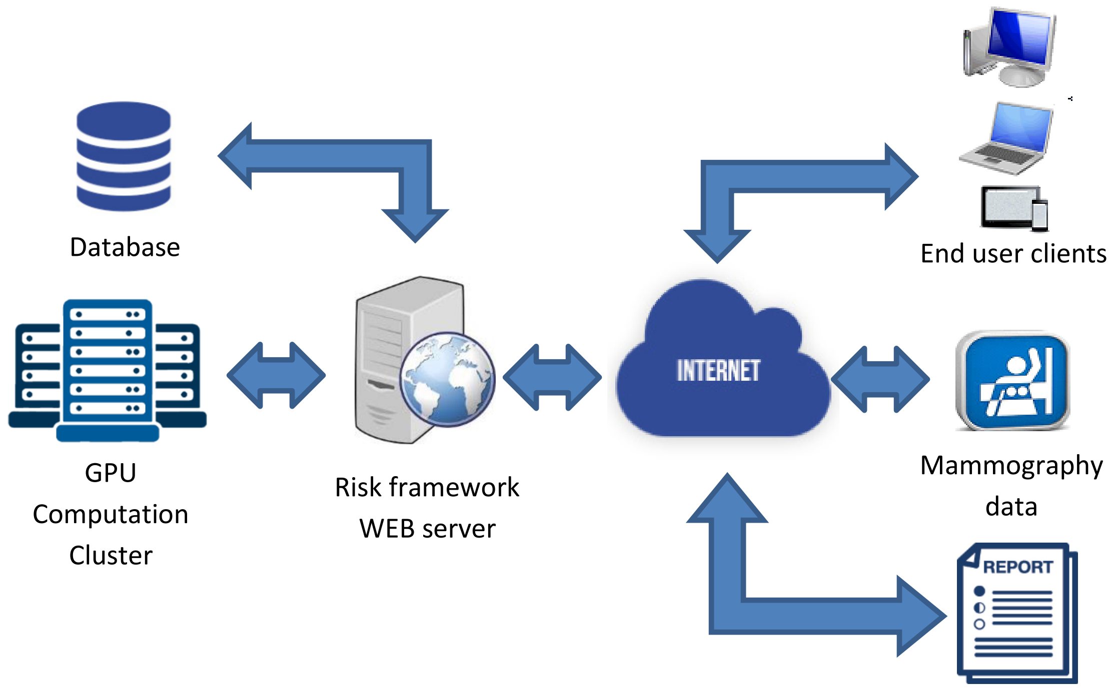
Table of Contents

1. Introduction

Nowadays, the standard methods for detection and diagnosis of breast cancer have several disadvantages, especially not being sophisticated enough, which sometimes lead to false-positive backs or even death. The idea of the following software module Bio Phantom Explorer is to generate a synthetic three-dimensional (3D) model of the breast, which is filled with geometrical primitives. The computational model is also used for the generation of X-Ray synthetic images. In x-ray medical imaging, the formation of the x-ray image is based on the absorption of the body tissues which absorption depends on both the density and the elemental composition of the tissues. The goal of the x-ray images is to screen and diagnose an injury or disease conditions. At the same time, however, the x-ray beam, which is used to produce the image, is harmful for the human body especially if the exposure to these rays is frequent. With the computer simulation and the use of synthetic models in virtual studies, unlimited tests and radiation could be generated which solves the problem with the need to use patients in these studies. Phantom Explorer also generates a 4D model of tumor formation, where the 4^{th} dimension is time. The tumor is later applied in the model and being also ray-traced and applied onto the x-ray picture.

The idea of anthropomorphic phantoms is to provide radiology with safe unlimited exposure. In this software solution they are used in a number of task including x-ray image assessment, simulation of malicious tissues, clinical commissioning and pre-clinical trials. Using the existing new diagnostic and interventional radiology technologies supports the clinical trials and it will be encouraged to develop skills to study and observe diseases and even death.

2. Overview



Bio Phantom Explorer is a modern system for generating 3D models of breast and breast tumor. The model resembles pressed female breast ready for a mammogram. Breast is represented with basic stereometric shapes. It consists of two semi-cylinders, the inner one filled with six different types of spheres. Based on Gang et al work's that shows that same volumes of different sized acrylic spheres provide an abstract representation of real biological tissues. That's why every one of those six types is equally-sized.

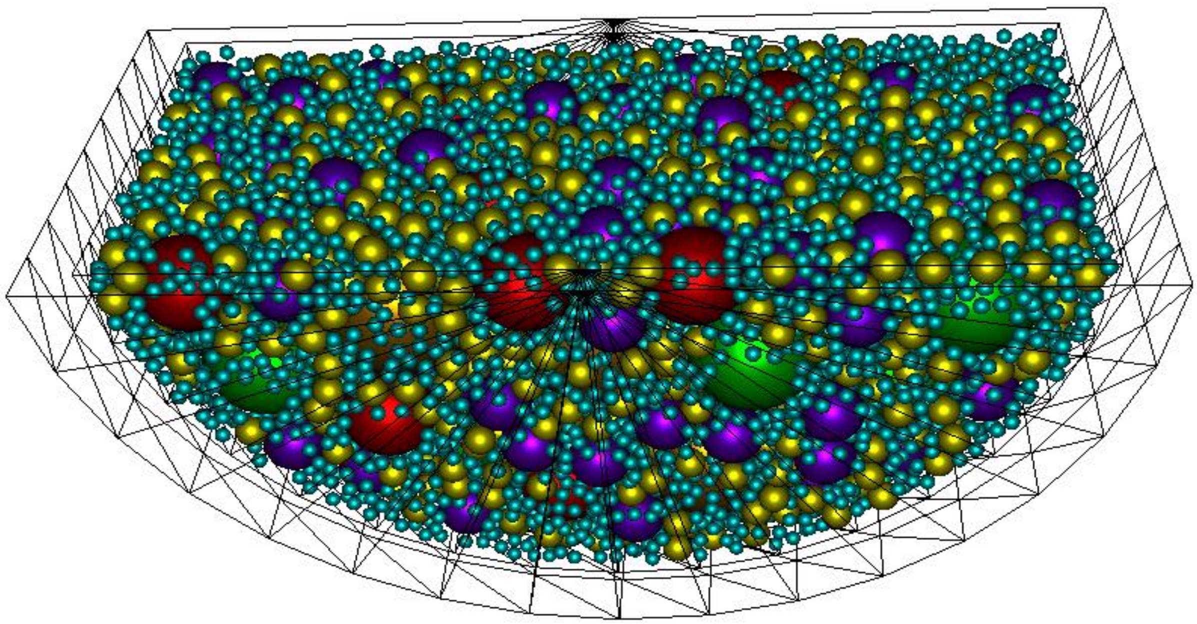
Description of summary architecture

Bio-Phantom Explorer is an application layer, which is an enclosure of RiskFramework(RFW). RFW is a software platform designed to develop and support a wide range of application cases, including:

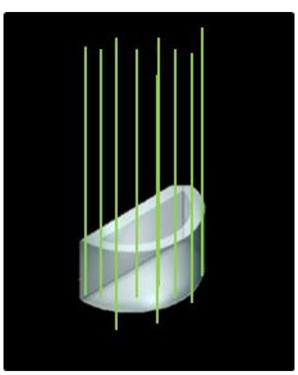
- Internet WEB Application Server - RFW runs in a WEB application server that is accessible via all standard WEB browsers (e.g. Internet Explorer, Mozilla Firefox, Google Chrome, Sea Monkey, Opera, Safari for PCs), as well as tablets and smart phones.
- Desktop Application - RFW runs on a separate computer.
- Intra Net Terminal Server - RFW runs in an application server within LAN.

All terminal servers, WEB browsers and desktop users are registered in RFW and can simultaneously access a common database server within a LAN, which ensures a transparent and cooperative work environment.

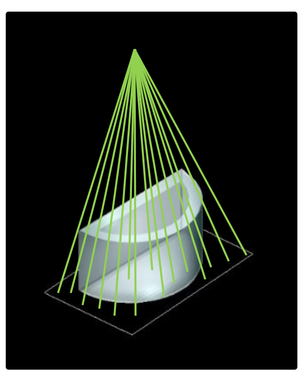
Risk Framework is designed be easily integrated into other existing systems, such as core banking systems, accounting systems, data provider, data warehouses, etc. Data from different application areas, sources and markets, that is stored in a data warehouse or taken from separate internal sources, can be directly accessed or imported into the RFW database via the standard importer. The data is then processed by modules of the RFW analytical engine and stored into different result tables within the database. Thereafter, advanced reporting tools, including Crystal reports, OLAP reports based on QlikView, and MS Exports, are applied and display printable results.



The system also makes 4 types of X-Ray images of the generated model, two of which are made simulating real world wave penetration. Those are Fan Beam and Rotation. There two types of beams used:



Parallel Beam



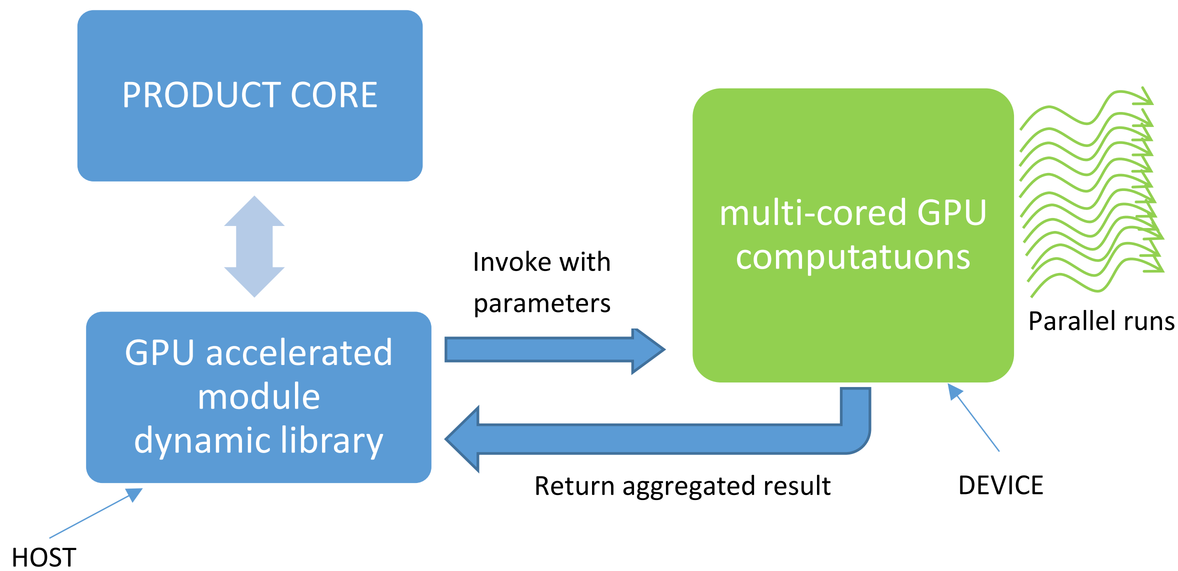
Fan Beam

Bio Phantom Explorer can generate a 4D tumor formation that could be applied into a model. For the creation of the abnormality is used Brownian Motion that represents the eternal movement of particles in nature.

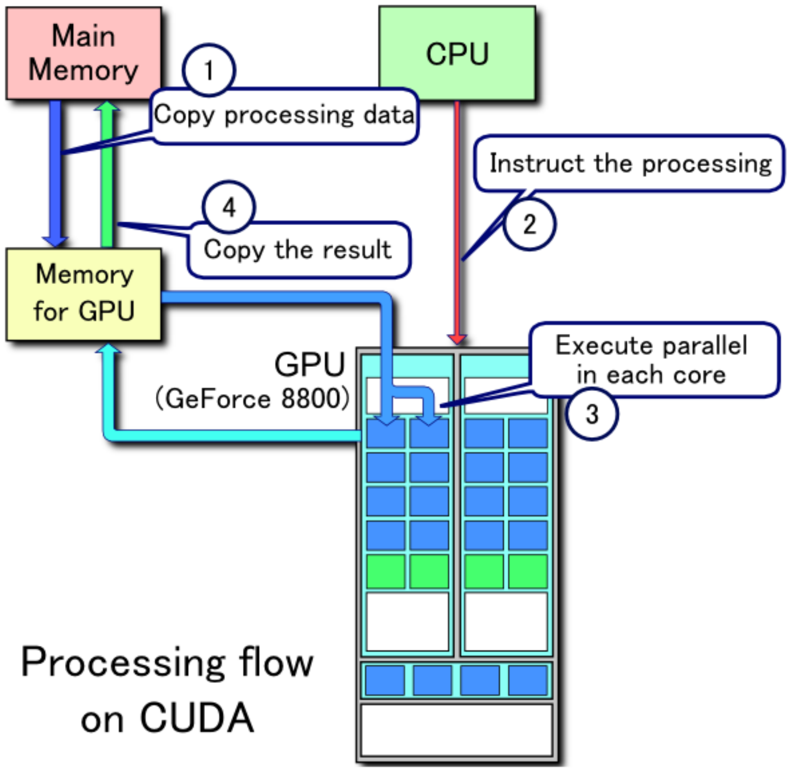
3. Used Technologies

CUDA

GPU is graphical processor unit. It presented as a "single-chip processor with integrated transform, lighting, triangle setup/clipping, and rendering engines”, but in finance we could use the potential of this resource to better financial computing. It’s made sense because faster pricing gains more revenue, more modeling gains less risk and maximizing resources gives more efficiency. The architecture of GPUs allows you to run more simulations that would increase the quality of your results. With more confidence in your data, you are able to offer tighter spread and gain competiveness. GPU even make it possible to run complex model, that were even impossible. You can obtain results of very complex models in real/near time, rather than overnight, and also provide deeper insight.



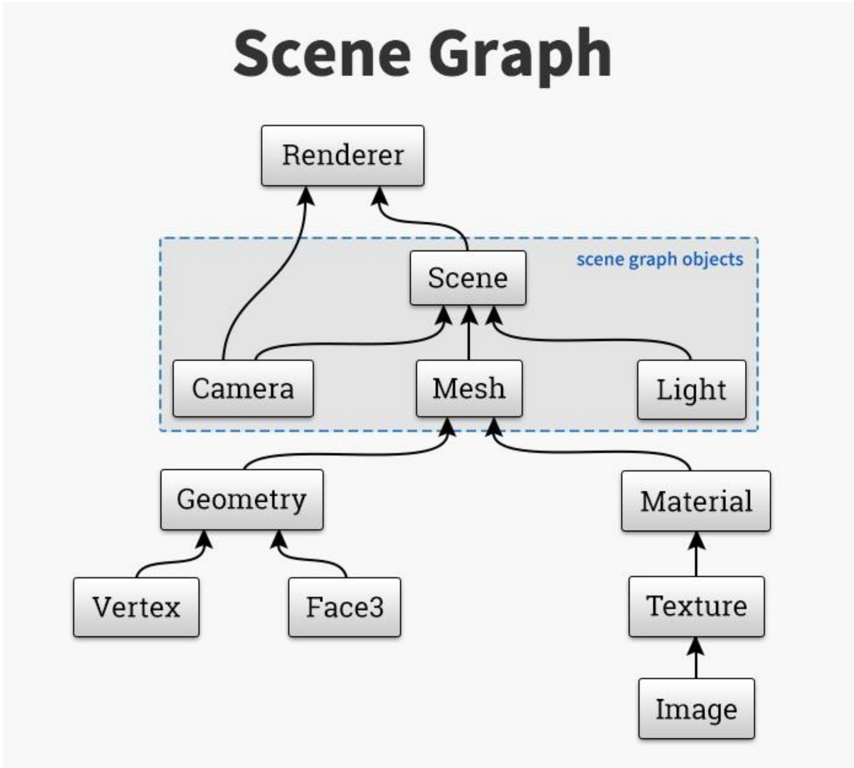
CUDA is a parallel computing platform and application programming interface (API) model created and maintained by Nvidia. It allows that the software to use a CUDA-enabled GPUs for general purpose tasks. The CUDA platform is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels.



Another advantage is multi-GPU support, theoretically provides as much more performance. For example, system with 8 same GPUs will provide 8 times faster than one. The access of data between CPU and GPU is much faster, because unified memory architecture. The GPU accelerated modules will provide at least 10 times faster calculation and it is quite possible to achieve 200 times speed up depending of applied parallel algorithm.

Three JS

Three.js is a cross-browser JavaScript library and Application Programming Interface (API) used to create and display animated 3D computer graphics in a web browser. Three.js uses WebGL. The source code is hosted in a repository on GitHub. Three.js allows the creation of Graphical Processing Unit (GPU)-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. This is possible due to the advent of WebGL. The main advantage is that is web based and it can be accessed from every time and almost every device, including personal computer, laptop, tablet, smartphone, etc.



For the filling of the model, simulation of the X-ray images and generation of the tumor formation is used the GPU(graphical processor unit).GPU accelerated modules are designed on 2 parts, host and device. Host is job for CPU like preparing the algorithm, main computing will be intergraded in device and will return the result. GPU acceleration is only for massively paralleled algorithms, for example use of at least 1000 threads.

4. Model description

Description of synthetic model

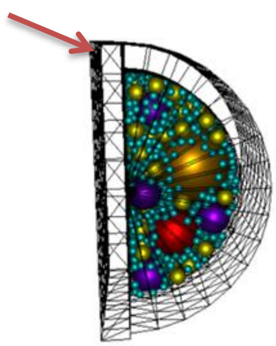
The fundamental part for performing diagnostic radiology and radiotherapy is the three dimensional object modeling. Every object is characterized by its geometry as well as mathematical attributes. The material properties include specific characteristic such as density, mass, weight, elemental composition, etc. The synthetic model is based on simple geometry, which is represented by simple geometrical primitives like sphere, cone, cube, cylinder, etc., and shape based, which is based on the boundaries of shapes by mathematical equations.

Representing tissue with spheres is a common practice in phantom synthesis. Most of the phantoms consists of spheres made of material that has properties similar to these of the breast tissue. For instance, the physical phantom of Cockmartin et al is based on Gang et al work's, which proposed that same volumes of different sized acrylic spheres provide an abstract representation of real biological tissues. That is why for the generation of model are used six types of spheres with different radii. In this paper, the physical container radii are 7.94, 6.35, 4.76, 3.17, 1.59 and 0.79 in physical measure, as well as the generated synthetic one

Development of 3D computer model of the human breast includes modelling of two semi-cylinders, consecutively filled with 6 different types of spheres. They are of different size and color, analogous to individual mammary glands. The generation of the 3D model is a heavy filling algorithm and thus is used Graphical Processor Unit (GPU) for it. Computing power which GPU provides is cheaper and faster, especially for task, which are suitable for massive parallelism. The used technology for the generation is CUDA, because it provides efficiency and development tools, which helps the overall progress of building and maintaining the module. Furthermore, with more detailed and large model, the conventional solution with CPU because the lack of efficient random number generator as well as the computation time for the generation.

Overall the descriptive object used in this library are:

- Point
  - X(float) – representing x coordinate in 3D coordinate system;
  - Y(float) – representing y coordinate in 3D coordinate system;
  - Z(float) – representing z coordinate in 3D coordinate system;
- Sphere
  - Center(point);
  - Radius(float);
- Line
  - Start(point) start point of the line
  - End(point) end point of the line
  - Distances (array of floats) – distance of the line in each type of descriptive type of tissues. Array is with constant length. The given types are for six types of spheres, water, skin and tumour
- Container(semi-cylinder)
  - Radius(float)
  - Height(float)
  - Closing(float) – size of the wall with is on the inside part of the container



  - Wall Thickness(float) – depth of the skin
- Plate – the image that is generated from the x-ray irradiating
  - Height(float) - physical size of the height of the image in real measurement (i.e. mm);
  - Width(float) - physical size of the width of the image in real measurement (i.e. mm);
  - Logical resolution by X and Y (unsigned integer) – the image, generated in pixels, abstraction of dots per inch
  - Distance(float) - distance from (0,0,0) to plate.

NOTE: All given physical measurements are in mm.

Database model

Each generated model is stored in the database and this are the columns for each of them. In RFW several phantoms can be stored in database, and later they can be easily edited. All the columns are shown below as well as the description for them.

Column name	Description
Object type	Type of described object (e.g. sphere, container wall etc.)
X	X coordinate
Y	Y coordinate
Z	Z coordinate
Hiperb	Hyperbolic coefficient of shape(only used for complex forms)
Rx	Stretching by X(for spheres this is radius)
Ry	Stretching by Y(for spheres this is radius)
Rz	Stretching by Z(for spheres this is radius)
H izk	Deviation of height
Euler angle	represents a rotation around the z axis
Thita	represents a rotation around the x' axis
Ksi	represents a rotation around the z" axis
Absorb coef	attenuation coefficient
Material(MC)	Description of complex compound(e.g. CaCO3, H2O, PMMA)
Number Mendeleev	If element is from simple compound this should the periodical number in the Mendeleev's table
Type(mc)	Type of compound
Grid	Special column 1
Type of grid	Special type
Grid	Special column 2
grid	Special column 3
Grid	Special column 4
Grid	Special column 5
Grid	Special column 6
Grid	Special column 7
Grid	Special column 8
Color	Decimal representation of #RRGGBB notation
Name	Custom name for the object

In addition, there is another table in each model, which is x-ray irradiation input parameters. Given columns are described below, as well as example data.

Column name	Description
Choice	Selected irradiation parameters
Energy	Energy of the beam in keV

H2O Att. Coef.	Attenuation coefficient of water for given energy(in most cases this is the material that is filled in container)
PMMA Att. Coef.	Attenuation coefficient of polymethylmethacrylate for given energy (in most cases this is the material of spheres)
Skin Att. Coef.	Attenuation coefficient of skin for given energy (in most cases this is corresponding to the wall)
CaCO3 Att. Coef.	Attenuation coefficient of tumor cells for given energy

It is important this table to be set, before processing the irradiation algorithm. Sample data is shown below. Best results are shown with energy equal to 0.0200.

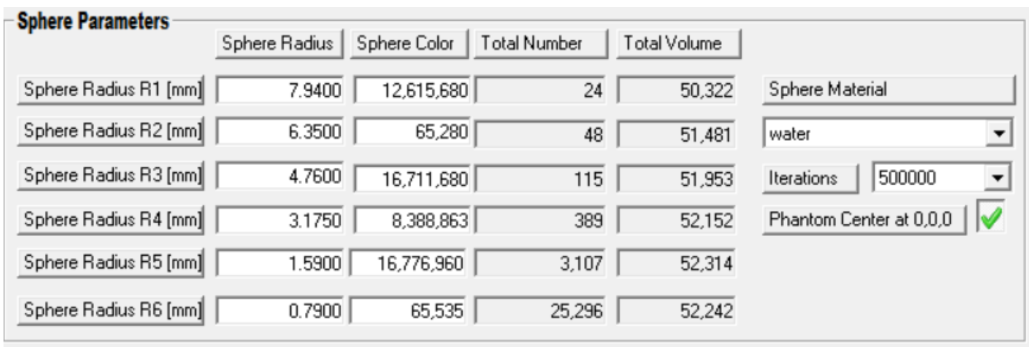
Number	Choice	Energy	H2O Att. Coef.	PMMA Att. Coef.	Skin Att. Coef.	Caco3 Att. Coefficient
1	0	0.001	4,077,000.00	2,796,000.00	3,517,000.00	441,500.00
2	0	0.0015	1,376,000.00	916,000.00	1,175,000.00	151,300.00
3	0	0.002	617,300.00	404,000.00	524,400.00	68,990.00
4	0	0.003	192,800.00	123,700.00	168,500.00	22,210.00
5	0	0.004	82,770.00	52,510.00	73,090.00	9,799.00
6	0	0.005	42,590.00	26,830.00	37,760.00	26,650.00
7	0	0.006	24,640.00	15,460.00	21,890.00	16,400.00
8	0	0.008	10,370.00	6,499.00	9,256.00	7,526.00
9	0	0.01	5,330.00	3,360.00	4,779.00	4,054.00
10	0	0.015	1,672.00	1,102.00	1,520.00	1,291.00
11	1	0.02	809.8	571.6	747.6	569.8
12	0	0.03	375.6	303.2	356.8	184.5
13	0	0.04	268.3	235	259.4	88.19
14	0	0.05	226.9	207.3	221.4	53.3
15	0	0.06	205.9	192.3	201.9	37.59
16	0	0.08	183.7	175	180.9	24.62
17	0	0.1	170.7	164	168.5	19.55
18	0	0.15	150.5	145.6	148.8	14.84
19	0	0.2	137	132.8	135.5	12.92
20	0	0.3	118.6	115.1	117.4	10.88
21	0	0.4	106.1	103	105	9.651
22	0	0.5	96.87	94.05	95.86	8.777
23	0	0.6	89.56	86.97	88.63	8.1
24	0	0.8	78.66	76.37	77.83	7.099
25	0	1	70.72	68.66	69.98	6.377
26	0	1.022	69.97	67.95	69.24	6.309

27	0	1.25	63.23	61.4	62.57	5.701
28	0	1.5	57.54	55.88	56.94	5.193
29	0	2	49.42	47.94	48.88	4.483
30	0	2.044	48.83	47.38	48.3	4.433
31	0	3	39.69	38.42	39.23	3.666
32	0	4	34.03	32.85	33.6	3.212
33	0	5	30.31	29.18	29.9	2.926
34	0	6	27.7	26.58	27.29	2.735
35	0	7	25.77	24.65	25.36	2.601
36	0	8	24.29	23.17	23.87	2.502
37	0	9	23.13	21.99	22.71	2.429
38	0	10	22.19	21.04	21.76	2.374
39	0	11	21.42	20.26	20.99	2.331
40	0	12	20.79	19.61	20.35	2.297
41	0	13	20.25	19.06	19.8	2.271
42	0	14	19.8	18.59	19.34	2.251
43	0	15	19.41	18.19	18.95	2.235
44	0	16	19.08	17.85	18.62	2.224
45	0	18	18.54	17.28	18.06	2.208
46	0	20	18.13	16.84	17.64	2.2
47	0	22	17.82	16.5	17.32	2.199
48	0	24	17.57	16.23	17.06	2.201
49	0	26	17.38	16.01	16.86	2.205
50	0	28	17.22	15.83	16.7	2.212
51	0	30	17.1	15.69	16.57	2.219
52	0	40	16.79	15.28	16.21	2.268
53	0	50	16.74	15.15	16.13	2.32
54	0	60	16.79	15.14	16.16	2.368
55	0	80	17.02	15.26	16.34	2.453
56	0	100	17.27	15.44	16.57	2.522
57	0	150	17.85	15.89	17.1	2.647
58	0	200	18.29	16.25	17.5	2.731
59	0	300	18.88	16.77	18.07	2.836
60	0	400	19.27	17.11	18.44	2.9
61	0	500	19.54	17.35	18.7	2.945
62	0	600	19.74	17.53	18.89	2.977
63	0	800	20.02	17.78	19.16	3.021

5. Model generation

Input

Most important parameters for model generation are containers radius, height, closing and wall thickness, where radius and height are refers to the inner semi-cylinder. Closing is the thickness of the wall that closes semi-cylinders. Wall thickness is the thickness of the surrounding wall that represents skin.

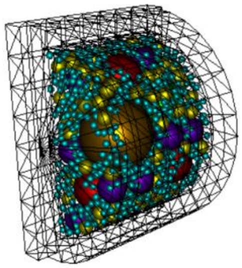


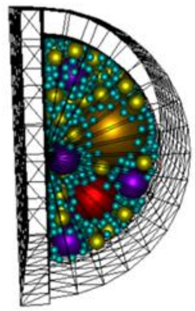
In the Spheres Parameters panel are given the radii of the six types spheres in millimeters.

There are two CUDA Parameters. First one (CUDA Blocks) is how many CUDA CORES to be used for the model generation. Second one is for the threads in the cores. It is recommended this number to be divided by 32 (for example 32, 64, 128).

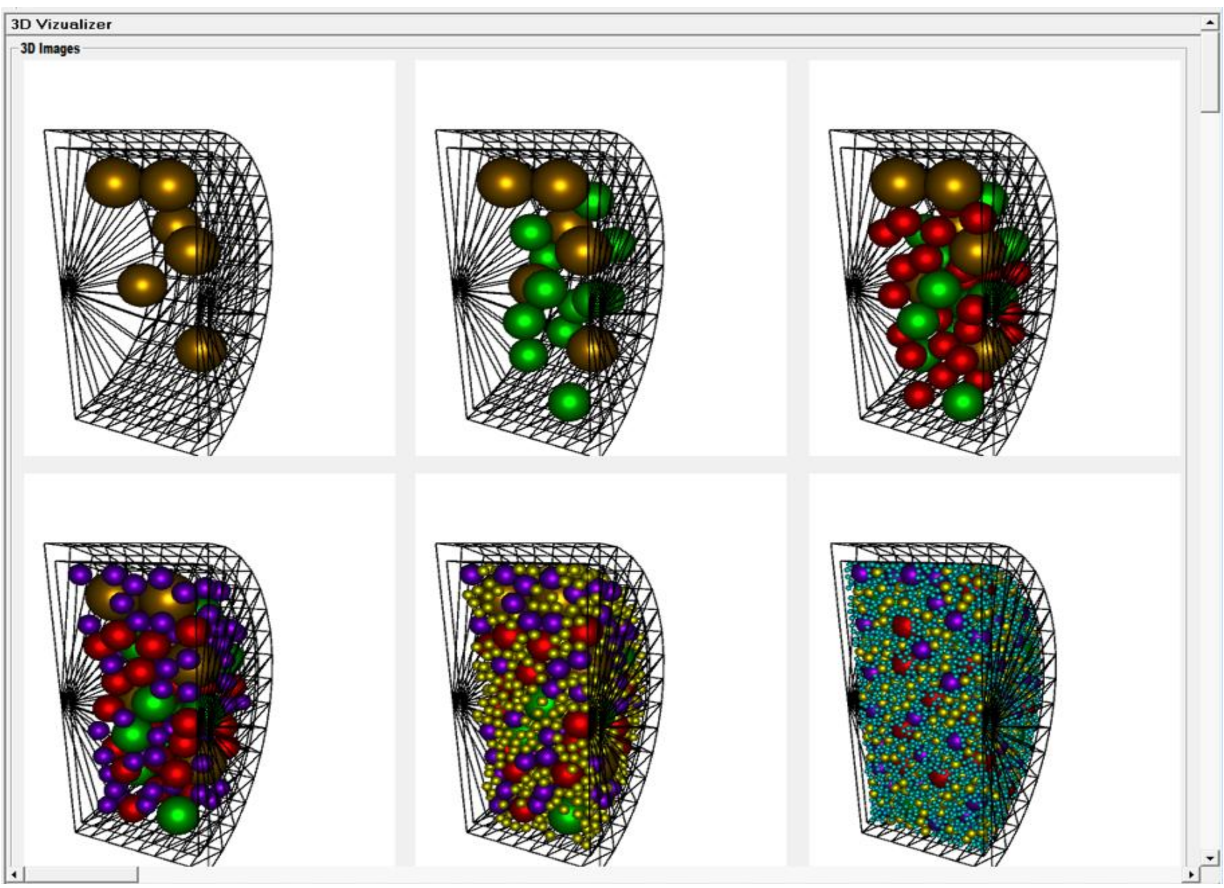
Output

The whole generated 3D model is visualized in the tab 3D Visualizer for Objects.





Filling of the model with spheres type by type is also shown and can be seen in the 3D Visualizer tab.



The data of the container is export in files for later use. It contains the radii of the spheres and the coordinates of the centers. There is also information about the wall thickness and the closing wall including the count of the spheres.

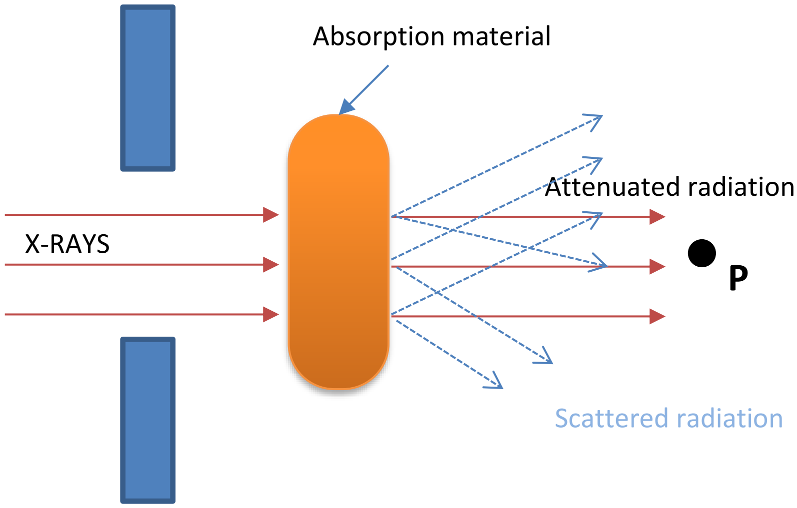
6. Irradiation of the phantom

The basic of x-ray is to make up a form of high-energy electromagnetic radiation. X-ray imaging creates pictures of the inside of human's body. The images show the parts of your body in different shades of black and white. This is because different tissues absorb different amounts of radiation. Calcium in bones absorbs x-rays the most, so bones look white. Fat and other soft tissues absorb less, and look gray. Air absorbs the least, so lungs look black. The most familiar use of x-rays is checking for broken bones, but x-rays are also used in other ways. For example, chest x-rays can spot pneumonia. In current topic x-rays are used to look for breast cancer.

The idea here is when an x-ray photon interacts with an atom in the absorption process, part of its energy E_{ph} is covered into kinetic energy of high speed practices, while part of radiated from the absorber as scattered radiation. Absorber dose is the energy deposed per unit mass of the medium. X-ray photons can be subjected to any of the four X-ray interaction types:

i. Coherent scattering
ii. Compton scattering
iii. Photoelectric absorption
iv. Pair production process

High energy particles are produced by the last three interaction and constitute the charge transport. The charged particles bear large number of elastic and inelastic collisions with the nuclei and the electrons until their fate. These include Moller and Bremsstrahlung interactions for the electrons and Bhabha and Annihilation interactions for the positrons. On figure bellow Is a diagram that illustrates how an attenuator of thickness 'x' reduces the number of photons reaching the any point of the detector. The removal of photons from a beam of x-rays as it passes through the material is called attenuation of x-rays. It is caused by both absorption and scattering of primary photons.

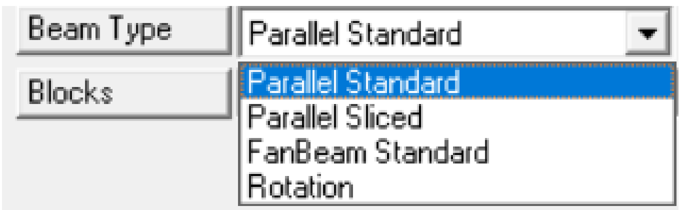


For the generation of X-Ray images are used several parameters

Number	Choice	Energy	H2O Att. Coef.	PMMA Att. Coef.	Skin Att. Coef.	Caco3 Att. (
1	☐	0.0100	5,330.0000	3,360.0000	4,779.0000	
2	☐	0.0150	1,672.0000	1,102.0000	1,520.0000	
3	☒	0.0200	809.8000	571.6000	747.6000	
4	☐	0.0300	375.6000	303.2000	356.8000	
5	☐	0.0400	268.3000	235.0000	259.4000	
6	☐	0.0500	226.9000	207.3000	221.4000	
7	☐	0.0600	205.9000	192.3000	201.9000	
8	☐	0.0800	183.7000	175.0000	180.9000	
9	☐	0.1000	170.7000	164.0000	168.5000	
10	☐	0.1500	150.5000	145.6000	148.8000	

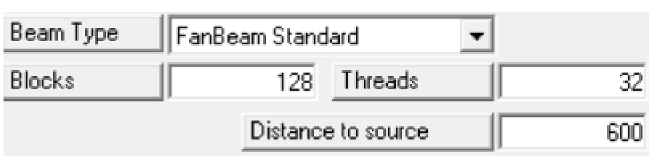
Where width and height stands for the physical measurements of the generated photo and Logical width/height are the pixels of the image. The bigger they are the more clear the generated image will be.

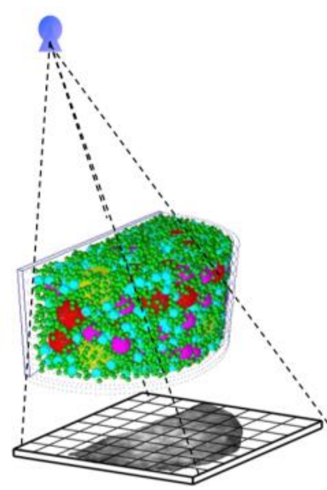
The ”Simple visualization” check button disables the opportunity of changes of the final images minimum and maximum values that excludes some parts of the image and turn theirs pixels black. It also disables the coloring option that turns the grey value in the corresponding HSV value. The distance that is given there is the distance between the center of the model and plate that stands for the end point of the x-ray waves. There are also boxes for CUDA parameters. The Beam Type combo box has all four options for irradiation. Each coming with its extra parameter/s except for the parallel. Then all of the rays are parallel which is not possible in



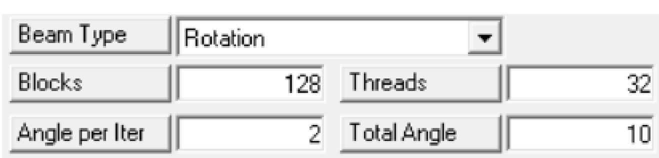
When simulation is done on slices the extra parameter stands for the count of the equal slices of the model.

are FanBeam irradiation copies the real world irradiation. In it all of the rays are coming from one point. The extra parameter is the distance from the center of the model to the source of the beam.





The last irradiation is Rotation. It uses fan beam. This type makes several images from a different angle.

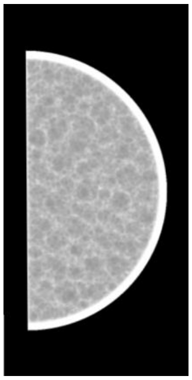


Here are needed the total angle of the arc of the source beams rotation and the difference in angles between the old and the new position of the source.

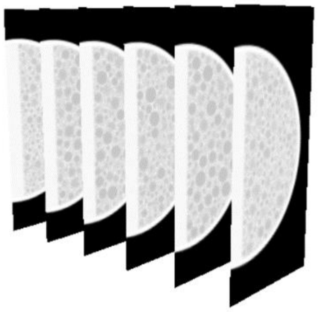
Results

Results of the irradiation are in the X-Ray photo panel.

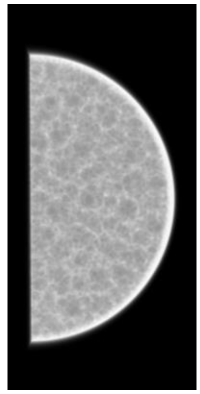
Parallel:



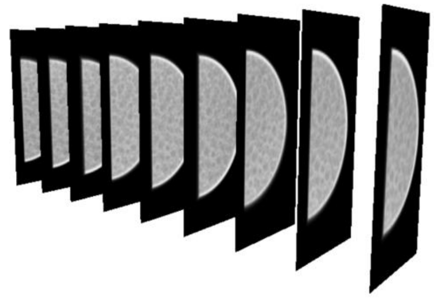
Sliced (6 slices):



Fan Beam:

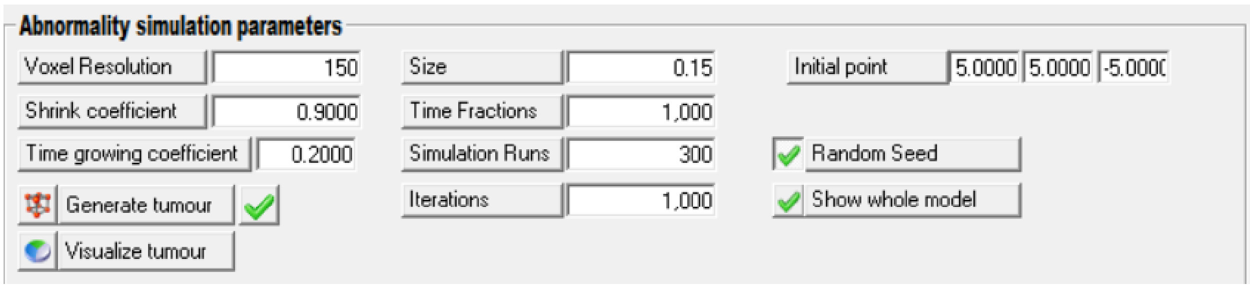


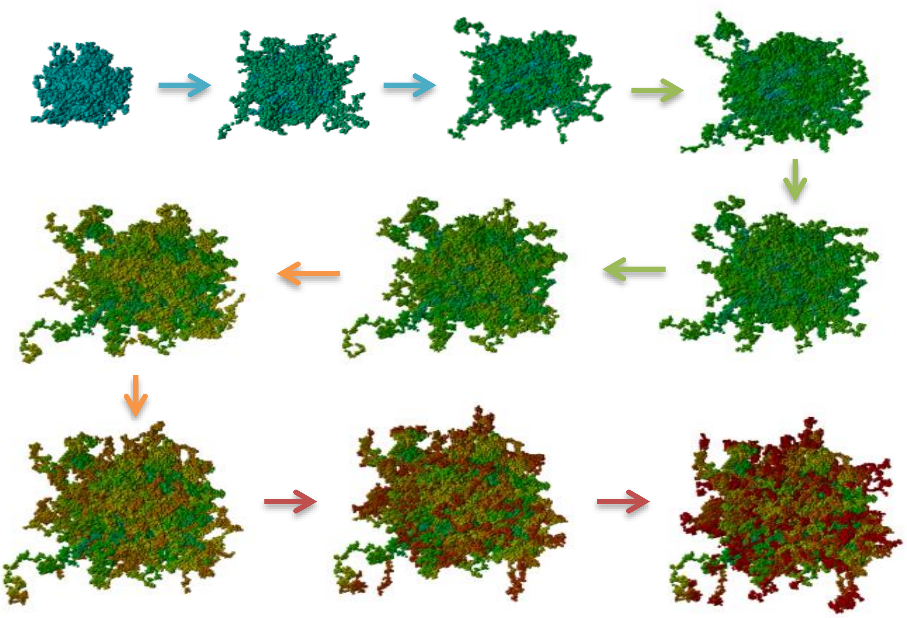
Rotation:



7. Abnormality

For the generation of the abnormality are used several parameters like the resolution of the model its physical size and physical initial point coordinates. Others parameters are important for the shape of the abnormality. If the random seed is not checked there will be generates the same abnormality.

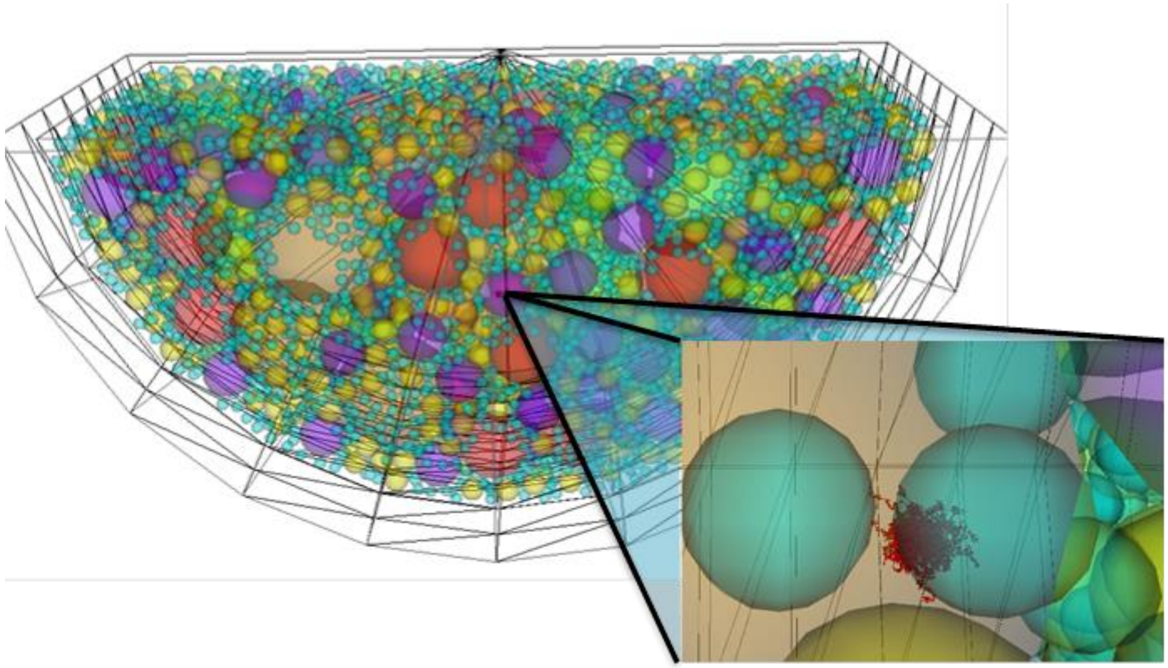




In result there is an abnormality that has its time periods in different color.

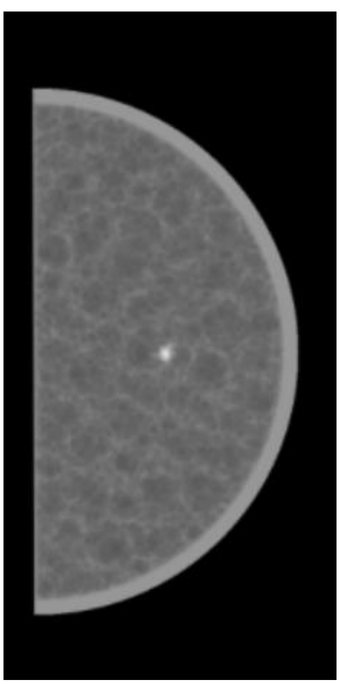
8. Applying the tumor into the phantom.

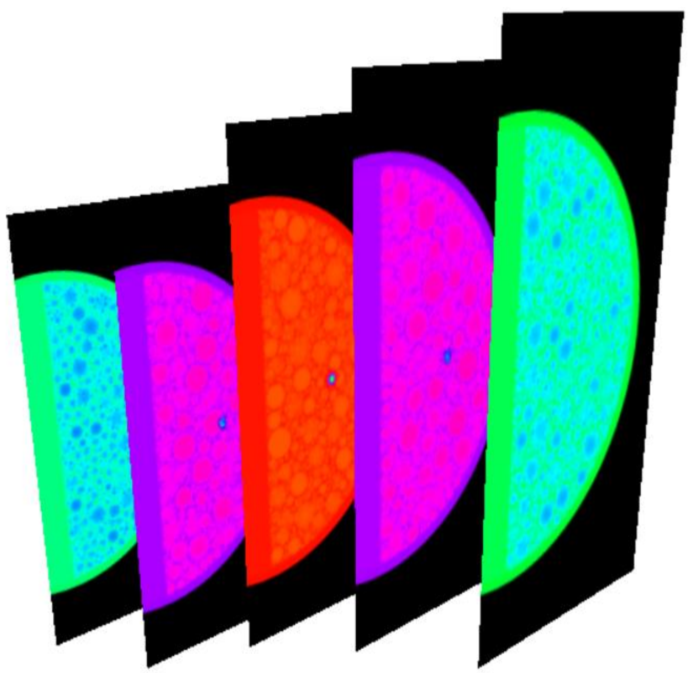
Applying the tumor in the 3d model



Applying the tumor on the X-Ray photos

Once an abnormality is generated it can be applied to any type of X-Ray image.





9. API Description

CudaSetUpParamers

Sets parameters regarding radii of the different type of spheres, container parameters and attenuation coefficient for each type of spheres. Also initialize the container instance in XRaySimulator object Parameters are:

- Radius(double) – radius of the semi-cylinder;
- Height(double) – height of the semi-cylinder;
- Closing(double) – depth of closing wall;
- Wall Thickness(double) – depth of the skin;
- arrRadius – array of 6 different radii for each type of descriptive spheres;
- arrCoefs – array of the attenuation coefficients for given sphere types.

Returns: 0 if all parameters are set.

LoadModel

Initialize the model with internal structure and copy the given data (array of Sphere structures). Parameters are:

- sph - sphere array pointer;
- size – number of elements in array.

Returns:

- 0 if data is set;
- -1 if input array in null pointer;
- -2 if size is not equal.

LoadRawModel

Initialize the model explicitly with arrays for each coordinate of sphere next to the radius and size. All four indices are corresponding to a sphere and data type is double. Parameters are:

- p_SpheresX – array of x coordinates for each sphere;
- p_SpheresY – array of y coordinates for each sphere;
- p_SpheresZ – array of Z coordinates for each sphere;
- p_SpheresRadius – array of radii for each sphere.

Returns:

- 0 if data is set;
- -1 if any input array in null pointer;
- -2 if size is not equal.

PreparePlate

Initialize plate parameters. It is mandatory to be initialized at least once before irradiating. Parameters are:

- Physical height of the plate(float) by default is in mm;
- Physical width of the plate(float) by default is in mm
- Logical height in pixel s– for result bitmap
- Logical width in pixels – for result bitmap
- Distance from zero – the absolute distance from point O(0,0,0) to plate – this is important only for fan beam irradiation type
- Caching directory – optional, path where both numeric and picture cache is stored, to prevent unnecessary calculations.

SetRadiationParams

Set the beam parameters. It is mandatory to be initialized before irradiation process. Parameters are:

- Intensity – Energy for the beam in kEV;
- attCoefH2O - Attenuation coefficient of water for given energy (in most cases this is the material that is filled in container);
- attCoefPMMA - Attenuation coefficient of polymethylmethacrylate for given energy (in most cases this is the material of spheres);
- attCoefSkin - Attenuation coefficient of skin for given energy (in most cases this is corresponding to the wall);
- attCoefCaCO3 - Attenuation coefficient of tumor cells for given energy

SimulateStandard

Simulates the simplest approach of irradiation – all lines are parallel, and calculation of attenuation is the simplest. Parameters are:

- launchBlocks – number of CUDA cores;
- launchThreads – number of CUDA threads;
- imageName – name of cached image, it is used for later analysis;
- outRes – image representation as numeric array of floats
- progress – functor for status bar.

Return status code:

- 0 for success;
- -1 for handled exception (getLastError function for detailed information)
- cudaError code for internal errors

SimulateSlicedOptimized

Simulates more sophisticated approach of irradiation – all lines are parallel, but radiation is on slices and model could be analyzed in range. Computation is optimized specially for slides simulation. Irradiation is computed partly for all sliced, and past objects are not computed twice. Parameters are:

- launchBlocks – number of CUDA cores;
- launchThreads – number of CUDA threads;
- imageName – name of cached image, it is used for later analysis;
- outRes – image representation as numeric array of floats
- progress – functor for status bar.
- Slices - number of slices

Return status code:

- 0 for success;
- -1 for handled exception (getLastError function for detailed information)
- cudaError code for internal errors

SimulateFanBeamRotation

Simulates more natural approach of irradiation – all beams are intersecting the model from different angles. Both distances are mandatory. Irradiation is computed once and the produced image is a little blurry. Parameters are:

- launchBlocks – number of CUDA cores;
- launchThreads – number of CUDA threads;
- imageName – name of cached image, it is used for later analysis;
- outRes – image representation as numeric array of floats
- progress – functor for status bar.
- sourceDist – distance of beam source from O(0,0,0)

Return status code:

- 0 for success;
- -1 for handled exception (getLastError function for detailed information)
- cudaError code for internal errors

SimulateFanBeamRotation

Simulates array of fan beam photos with rotating the beam source – all beams are intersecting the model from different angles. Both distances here is mandatory. Irradiation is computed as many times depending the angles and the produced image is a little blurry. Parameters are:

- launchBlocks – number of CUDA cores;

- launchThreads – number of CUDA threads;
- imageName – name of cached image, it is used for later analysis;
- outRes – image representation as numeric array of floats
- progress – functor for status bar.
- sourceDist – distance of beam source from O(0,0,0);
- anglePerIter – angle per iteration in degrees, this mean the movement of the source by x or y
- totalAngle – the arc of rotation, where the bisectrix is sitting of vector Oz

Return status code:

- 0 for success;
- -1 for handled exception (getLastError function for detailed information)
- cudaError code for internal errors

TerminateSimulation

Terminate the irradiation process via signals.

getLastError

Returns last exception description. If error is unexpected, a dmp file is generated, and can be traced in visual studio.

simulateAbnm

Generates the tumor tissue based on Brownian motion into the model. Generated tumor is voxelized. Parameters are:

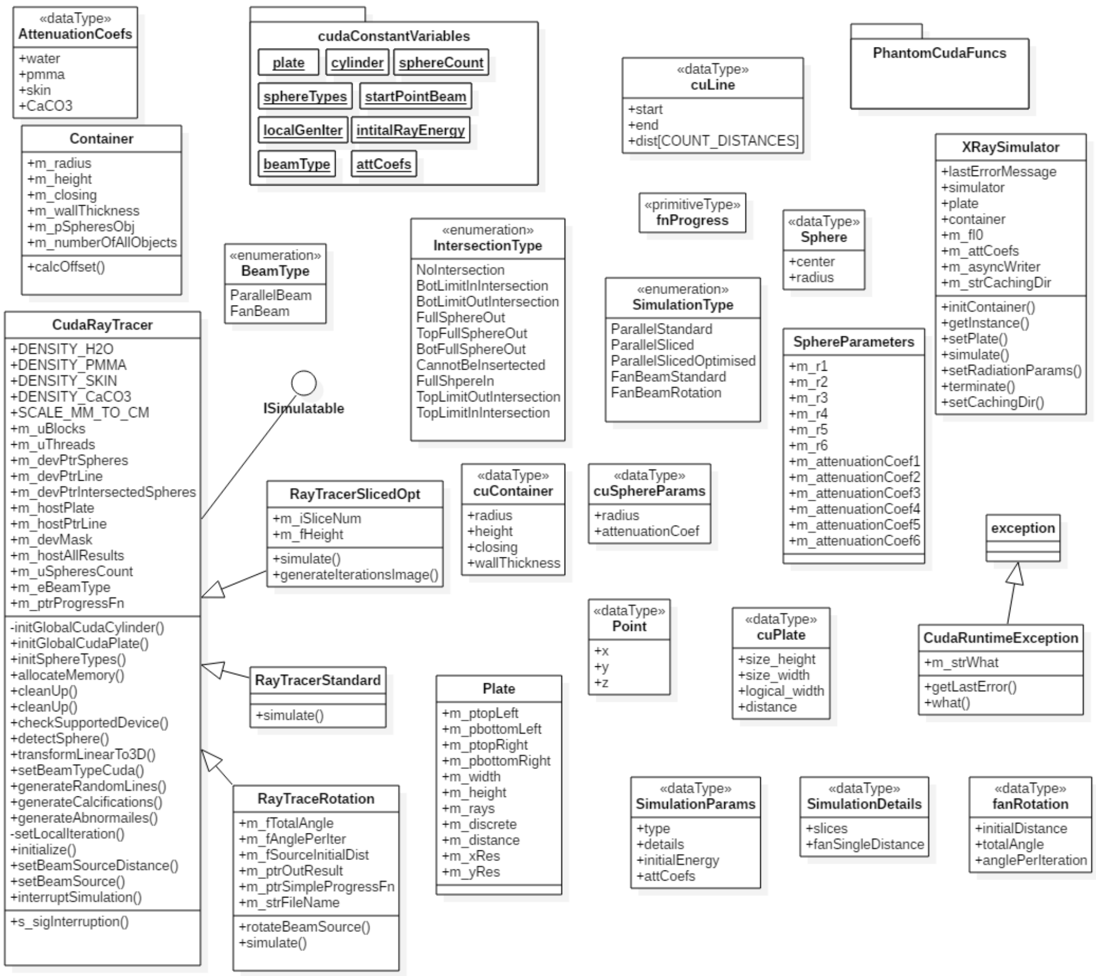
- p_VoxelRes – number of voxels(cubes), which describes the tumor
- p_ShrinkCoef – coefficient of determining the shrinkness of the tumor, value should be between 0.0 and 1.0, where the bigger is the coefficient, the tumor is more like a ball.
- p_timeCoef – value for distance between the generated voxels
- p_simulationIters – iteration of the simulation, determinates the length of generated paths
- p_simulationRuns – determinates the number of paths
- p_timeItersView – numbers of growing checkpoints
- p_fScaleCoef – ratio coefficient of physical to voxel
- p_results – numeric representation of xray model image
- p_initialPointArr – array of coordinates for initial point (x,y,z)
- p_attenuationCoef – attenuation coefficient for tumor
- p_autoSeed – seed of random numbers for generation process
- slices – only used for sliced irradiation of tumor
- p_fileName – image of cancer
- p_simType – type of irradiation, mandatory for fan beam rotation

Return status code:

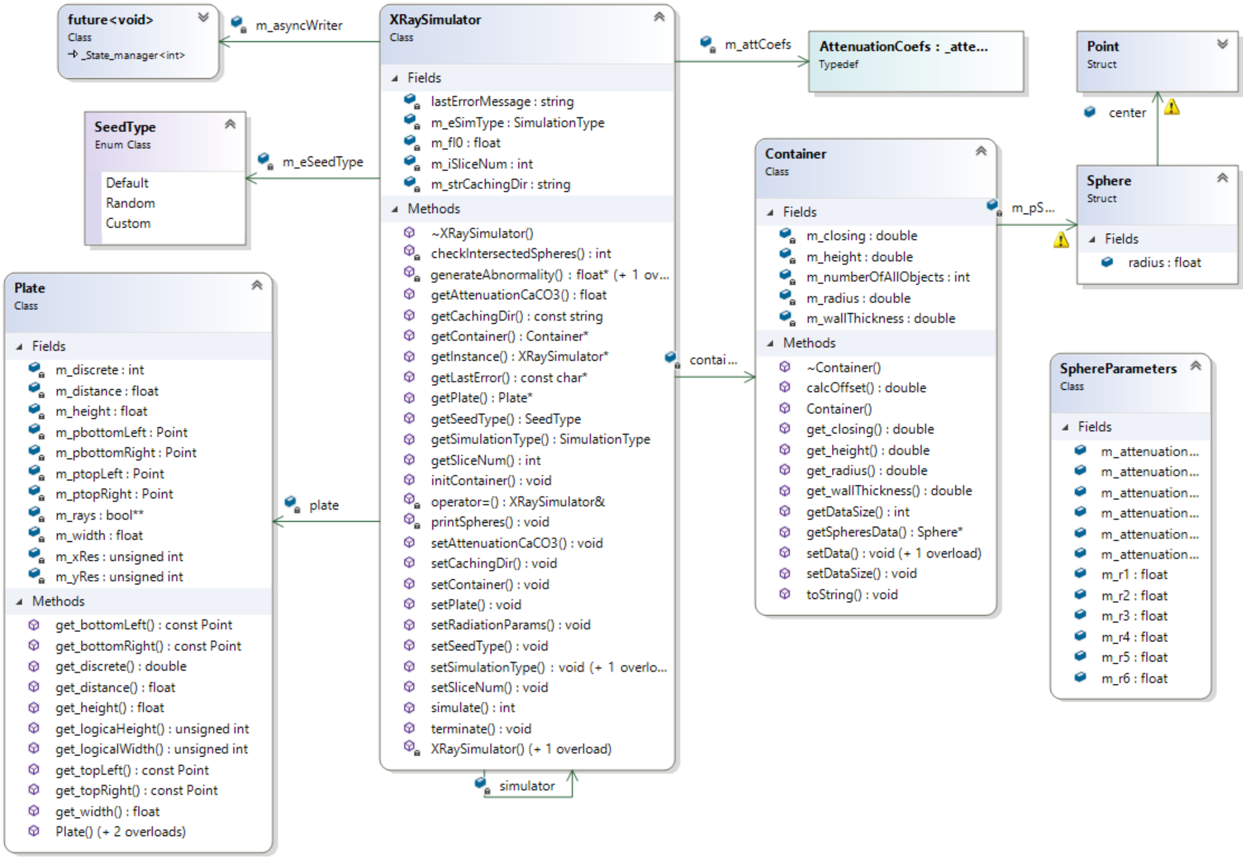
- 0 for success;
- -1 for handled exception (getLastError function for detailed information)
- cudaError code for internal errors

10. Irradiation Library Architecture

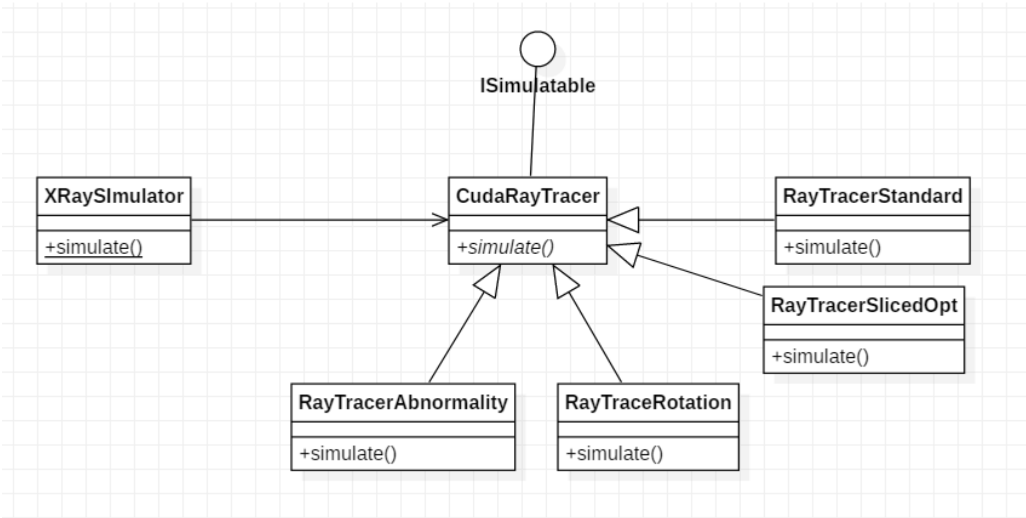
The overall architecture contains a number of controllers and models for computation algorithm. The main mathematics logic is placed in PhantomCudaFuns. Functions are described below. The library class diagram is shown below:



The entry points of the library are handled by the X-Ray Simulator. Scheme of the architecture is shown bellow.



Depending on the input parameters it creates different instance of the Ray Tracer to simulate the irradiation. This object follows singleton pattern, which means only one instance is available for whole object. The idea of X-ray simulator object is to manage the whole process and to store model data temporary. Furthermore, it acts like facade for all data source, which means that class manages creation and modification of all model classes (i.e. spheres, plate, container, etc. In addition, this is the object which is accessed from API. The method which produce the image is simulate. In this method an object from type CudaRayTracer, which is the controller class for the irradiation process. This class accomplish the connection between DLL and GPU computations. The methods are described in namespace 'PhantomCudaFuncs'. Methods in this namespace are separated on 2 types: global which is accessed from CudaRayTracer and device functions, which can be accessed internally. In addition, this objects set and operates the GPU memory and all objects including global memory and parameters.



Most important global functions in this namespace are:

- calculateAttenuationCoef

computes the attenuation coefficient for line and return the result via parameter, because connection between host and device is only via pointer parameters and here the pattern is followed

- getDistance

debug adapter functions between device code, which would be used in release as well as global functions, which provide key data to determine if that functionality works correct

- rayTrace

this method should summarize following functionalities
(1) generate lines
(2) get intersected spheres and its distance of line
(3) calculate distance in water, wall and space

(4) calculate attenuation coef

- findIndex

calculates index to be set in host memory and computed index is transformed to linear Yi*X + Xi

- generateDebugLines

moved out of debug, because for partly tracing, lines are generated once and then trace in depth

- rayTraceDN

Entry point for sliced ray tracing.

- generateBrownianMotionTumor

Generates tumor based on Brownian motion, model is stored as 3d voxel matrix

- calculateAbnormalityInSlice

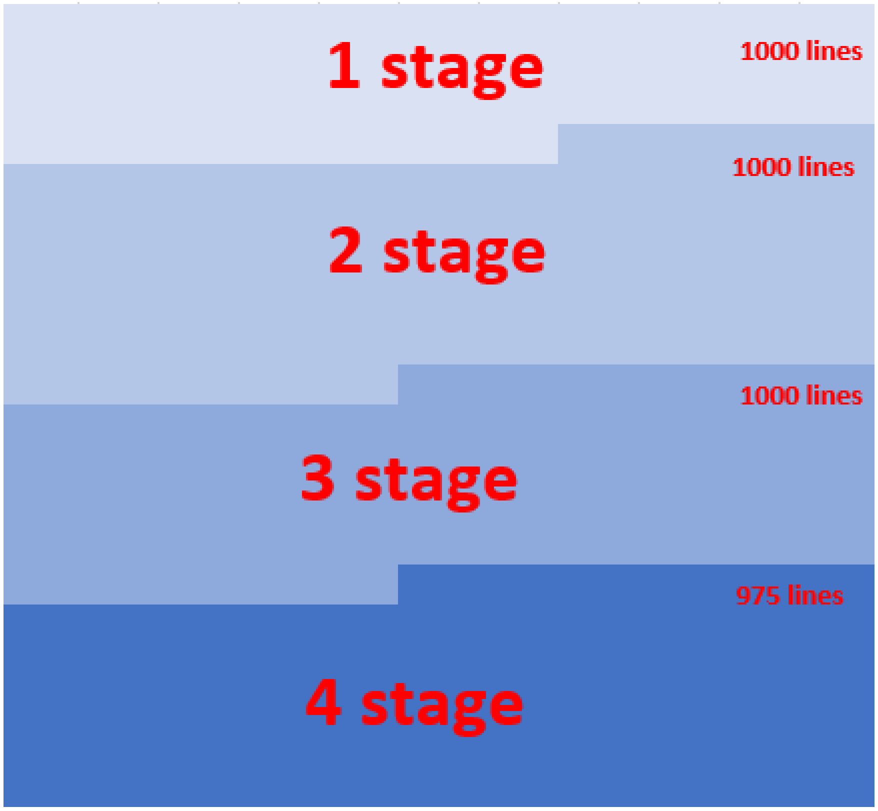
calculates distance in abnormality for sliced simulation

Device functions for CUDA are:

- getSphereType – return type of sphere depending on radius
- getIntersectionPoint – computes the intersection point between line and plate
- intersectCos – computes the distances of all intersected spheres for given line
- commonCaseIntersection – checks if line intersect both bases of the semi-cylinder
- calcDistanceInSpace – computes distance in space(water) in the semi cylinder
- CheckPointC -check if Point C (this is intersection point with cylinder is point of container(half cylinder from h/2 to -h/2))
- findEdgePoint- checks the edge point by given 2 points of the line, if the line intersect wall of the cylinder
- calcDistanceInCylinder - calculates distance of lines in cylinder
- calcDistanceInWater - calculates distance in water in cylinder
- calcDistanceInWall - calculates distance in wall(part between both semi cylinders)
- getSpheres - calculates distance that following ray(line) intersects particular spheres
- generateParallelLines - generates parallel lines -> each pixel corresponds to a line
- generateSequentialFanBeam - generates pyramid shaped beam -> each pixel corresponds to a line
- linePointIntersection - detects if line cross point
- arrayIndex - 3d get index helper

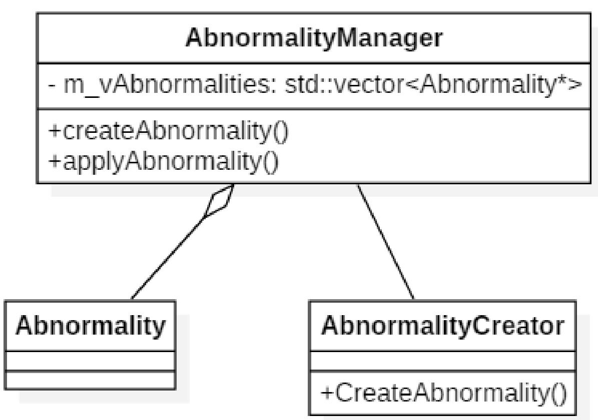
The namespace cudaConstantVariables is used for storage of shared variables, they are:

- cylinder
- plate
- count of spheres in model
- types of spheres – spheres can be with different type, different material and respectively different density
- start point of the beam – point of the beam source, it is used only for fan beam irradiation
- local iterator – it is used for step iterator, because all lines are processed party for the image, it means the the image is generated on stages, example of the stages is shown bellow



- initial beam energy
- beam type – it can be fan beam or standard
- attenuation coefficients for materials

For parallel and fan irradiation is called the Ray Tracer Standard. Rotation is handled by Ray Tracer Rotation and similarly slicing by Ray Tracer Sliced.

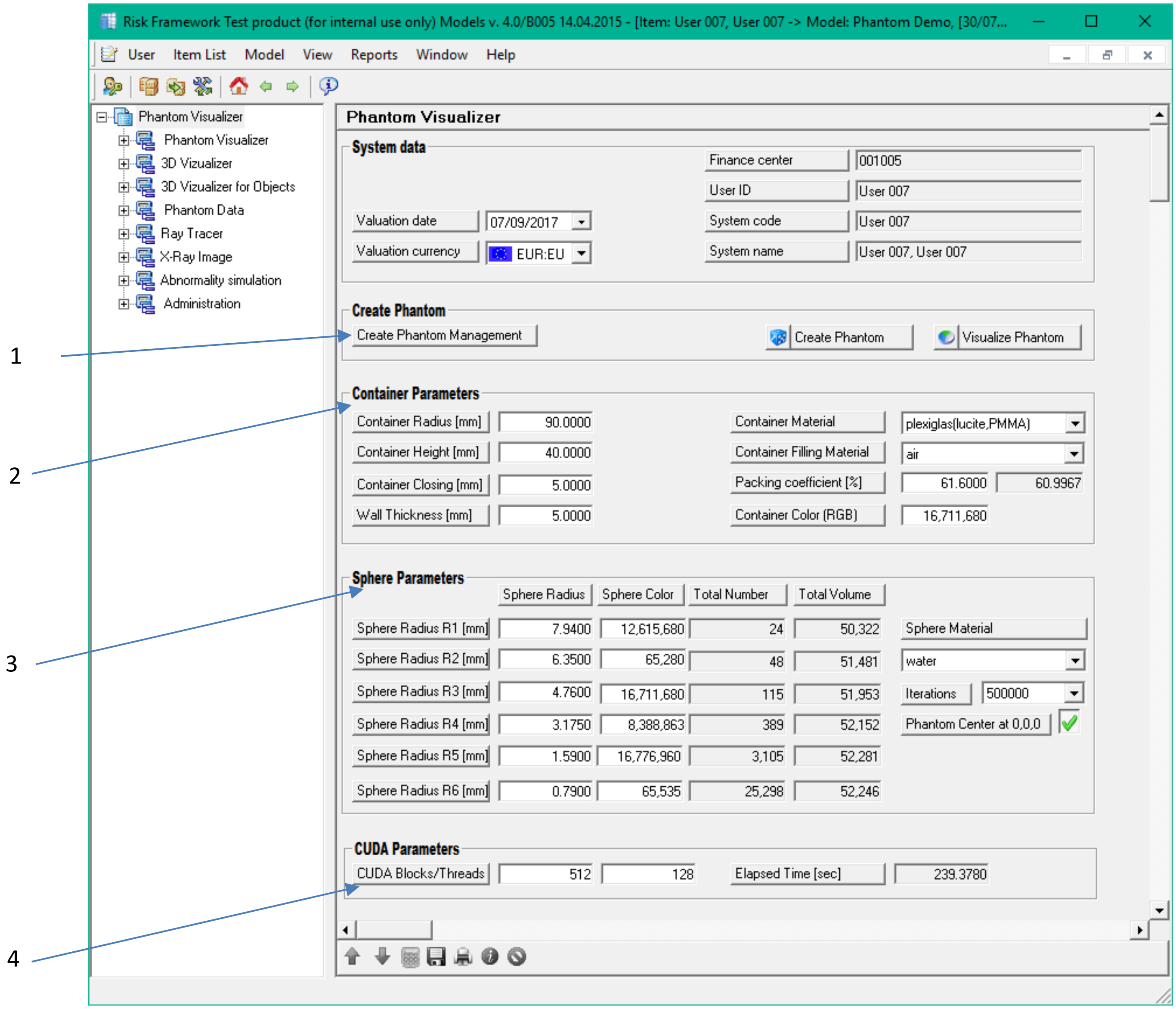


Abnormality Manager is responsible for the tumors. It contains all the generated tumors as a member and has the functionality to create more or to apply then into the models or the X-Ray photos. Abnormalities are created from Abnormality Creator that is called in the manager when the createAbnormality function has been invoked. They are being applied according to the type of irradiation with the applyAbnormality function.

This library also has custom exception that inherits exception from the standard namespace. There is also Image Generator that generated bitmap images from raw data in gray scale of colorful.

11. User guide

Phantom Visualizer



Create Phantom

This panel contains 2 buttons:

- Create Phantom – generates phantom with the given parameters
- Visualize Phantom - visualize the last generated phantom

Container Parameters

Most important parameters for model generation are containers radius, height, closing and wall thickness, where radius and height refers to the inner semi-cylinder. Closing is the thickness of the wall that closes semi-cylinders. Wall thickness is the thickness of the surrounding wall that represents skin. There are also combo boxes where the user can choose the material for the cylinder and its filling material. Parameter Packing coefficient stands for the percentage of the container volume that is filled with spheres.

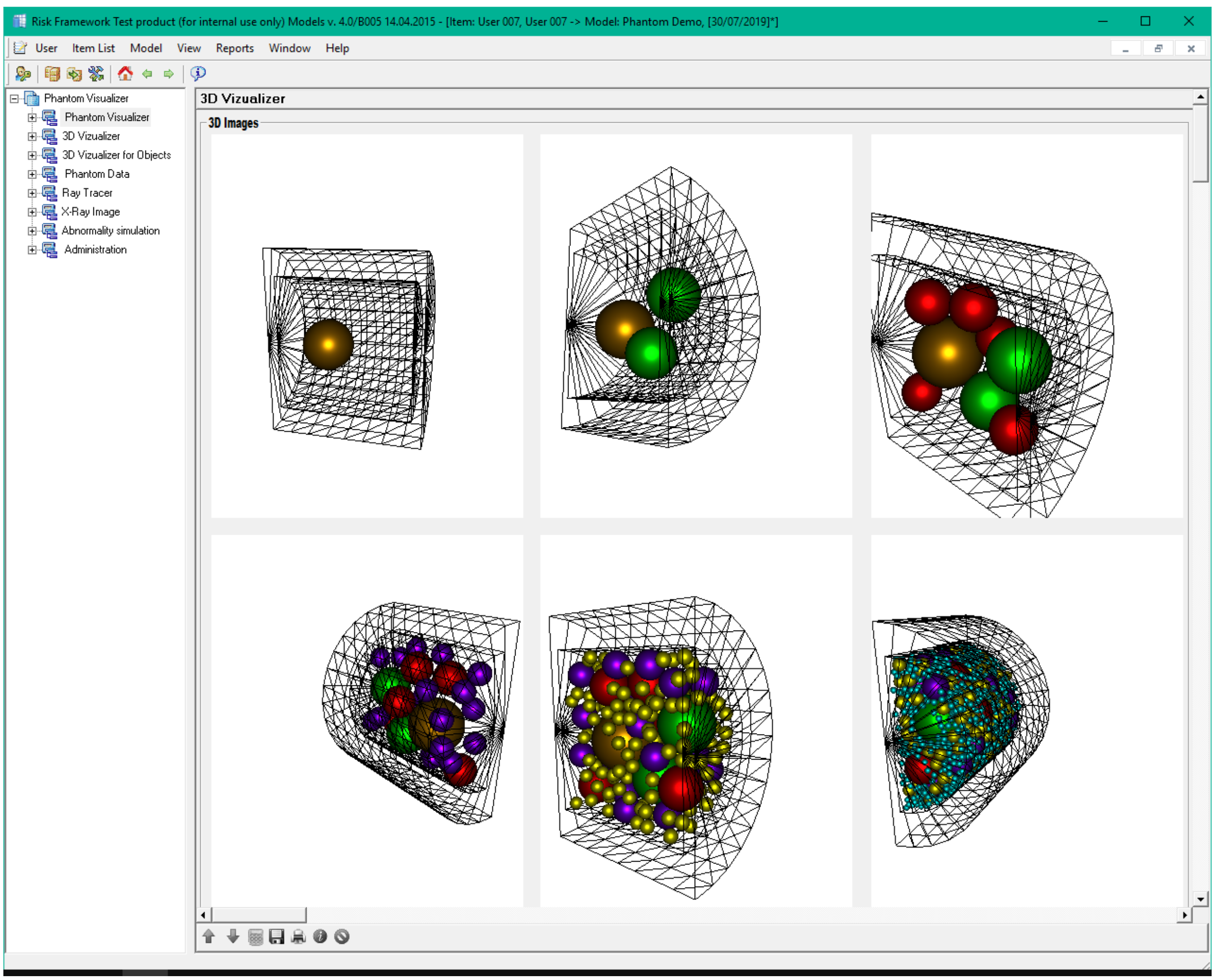
Sphere parameters

In the Spheres Parameters panel are given the radii of the six types spheres in millimeters and theirs corresponding colors in model that is later shown in the next two tabs. Here the user can choose which is the material of the spheres and how many runs (iterations) to be done for the filling.

CUDA Parameters

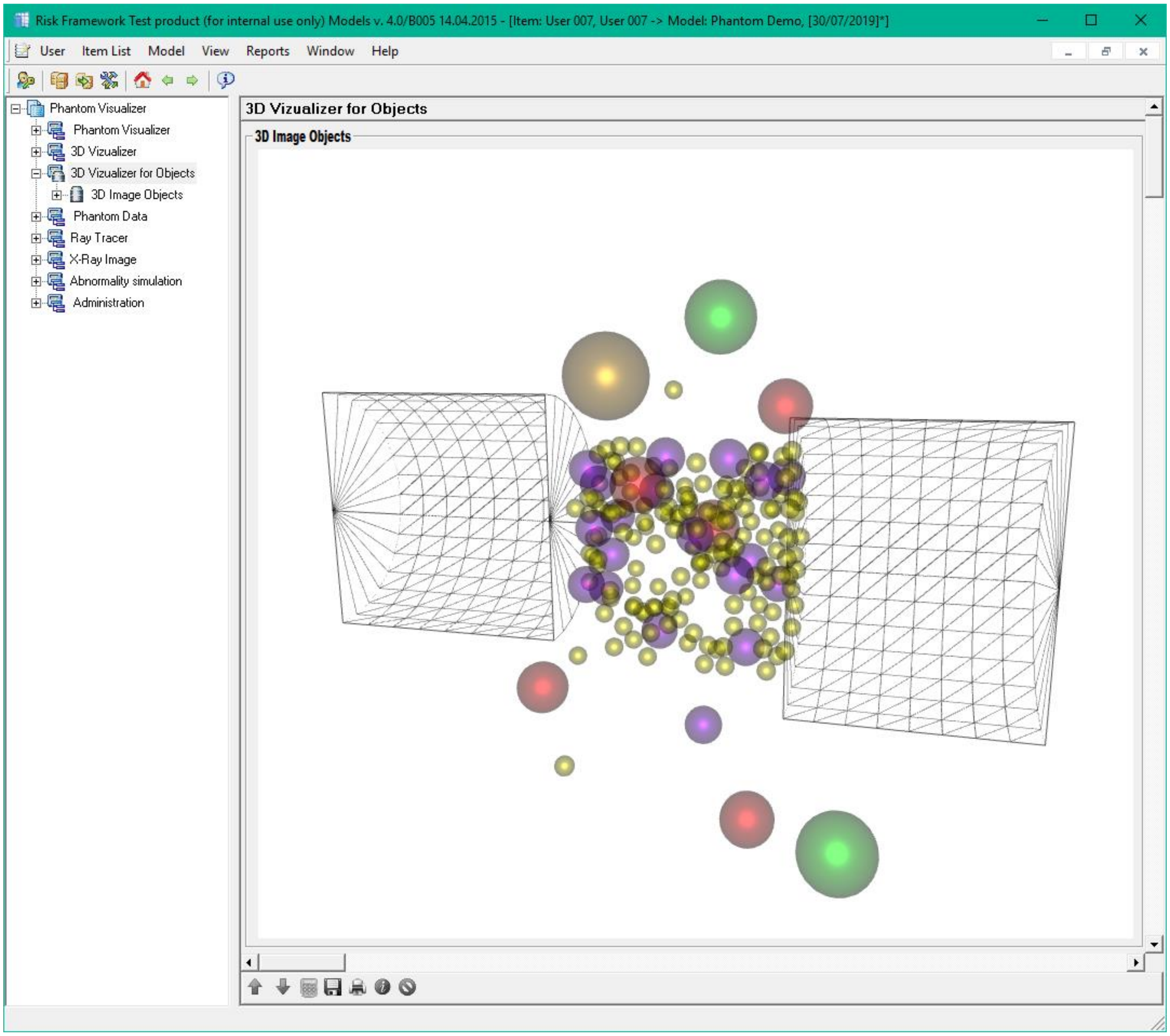
There are two CUDA Parameters. First one (CUDA Blocks) is how many CUDA CORES to be used for the model generation. Second one is for the threads in the cores. It is recommended this number to be divided by 32 (for example 32, 64, 128).

3D Vizualizer



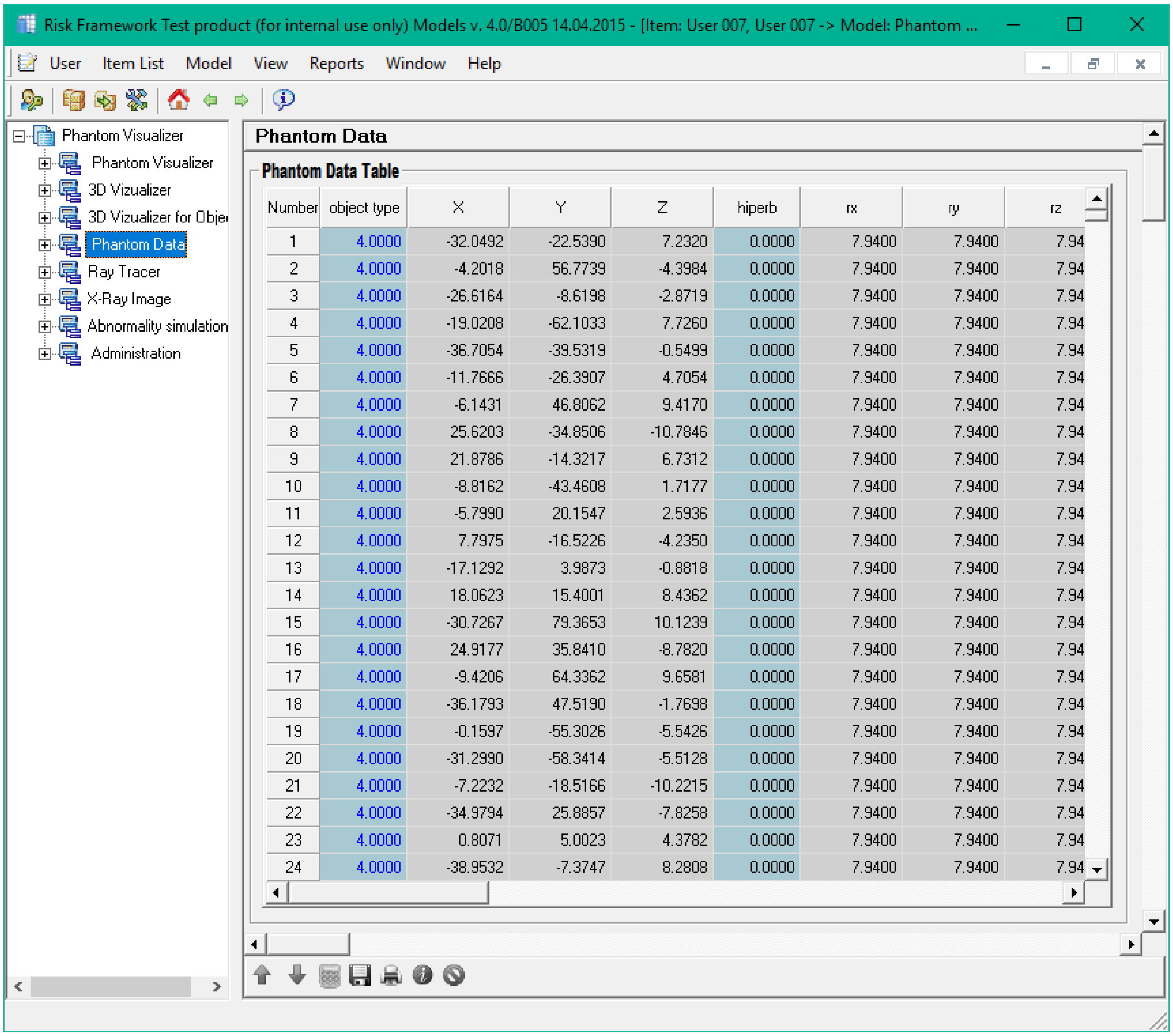
In this panel is shown the gradual filling of the container with spheres. There is a frame for every type spheres. The last frame contains the whole model.

3D Vizualizer for Objects



In this panel placed the slightly transparent model that can be unpacked and every single sphere can be moved around as well as the semi-cylinders.

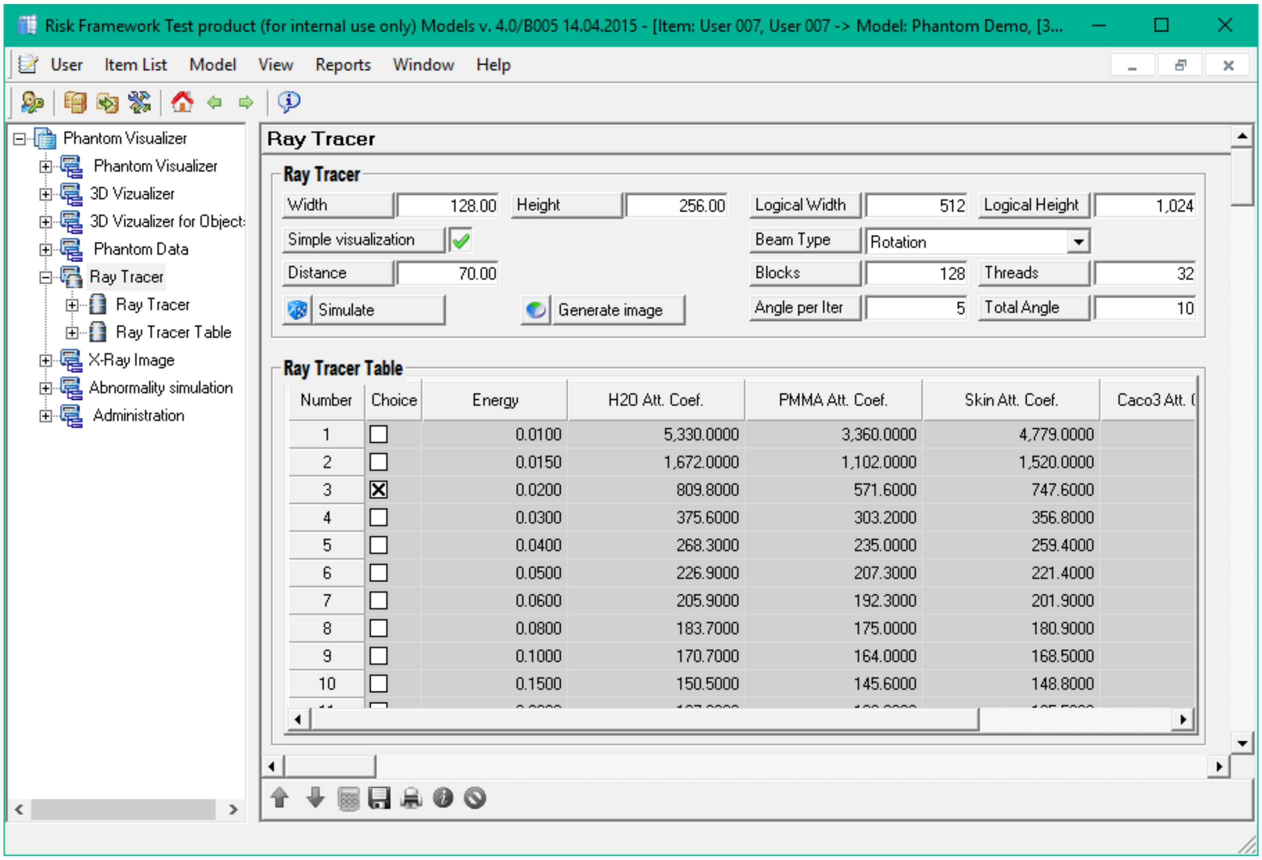
Phantom Data



Number	object type	X	Y	Z	hiperb	rx	ry	rz
1	4.0000	-32.0492	-22.5390	7.2320	0.0000	7.9400	7.9400	7.94
2	4.0000	-4.2018	56.7739	-4.3984	0.0000	7.9400	7.9400	7.94
3	4.0000	-26.6164	-8.6198	-2.8719	0.0000	7.9400	7.9400	7.94
4	4.0000	-19.0208	-62.1033	7.7260	0.0000	7.9400	7.9400	7.94
5	4.0000	-36.7054	-39.5319	-0.5499	0.0000	7.9400	7.9400	7.94
6	4.0000	-11.7666	-26.3907	4.7054	0.0000	7.9400	7.9400	7.94
7	4.0000	-6.1431	46.8062	9.4170	0.0000	7.9400	7.9400	7.94
8	4.0000	25.6203	-34.8506	-10.7846	0.0000	7.9400	7.9400	7.94
9	4.0000	21.8786	-14.3217	6.7312	0.0000	7.9400	7.9400	7.94
10	4.0000	-8.8162	-43.4608	1.7177	0.0000	7.9400	7.9400	7.94
11	4.0000	-5.7990	20.1547	2.5936	0.0000	7.9400	7.9400	7.94
12	4.0000	7.7975	-16.5226	-4.2350	0.0000	7.9400	7.9400	7.94
13	4.0000	-17.1292	3.9873	-0.8818	0.0000	7.9400	7.9400	7.94
14	4.0000	18.0623	15.4001	8.4362	0.0000	7.9400	7.9400	7.94
15	4.0000	-30.7267	79.3653	10.1239	0.0000	7.9400	7.9400	7.94
16	4.0000	24.9177	35.8410	-8.7820	0.0000	7.9400	7.9400	7.94
17	4.0000	-9.4206	64.3362	9.6581	0.0000	7.9400	7.9400	7.94
18	4.0000	-36.1793	47.5190	-1.7698	0.0000	7.9400	7.9400	7.94
19	4.0000	-0.1597	-55.3026	-5.5426	0.0000	7.9400	7.9400	7.94
20	4.0000	-31.2990	-58.3414	-5.5128	0.0000	7.9400	7.9400	7.94
21	4.0000	-7.2232	-18.5166	-10.2215	0.0000	7.9400	7.9400	7.94
22	4.0000	-34.9794	25.8857	-7.8258	0.0000	7.9400	7.9400	7.94
23	4.0000	0.8071	5.0023	4.3782	0.0000	7.9400	7.9400	7.94
24	4.0000	-38.9532	-7.3747	8.2808	0.0000	7.9400	7.9400	7.94

The previous table contains all the sphere parameters like position of their canter, radius ect.

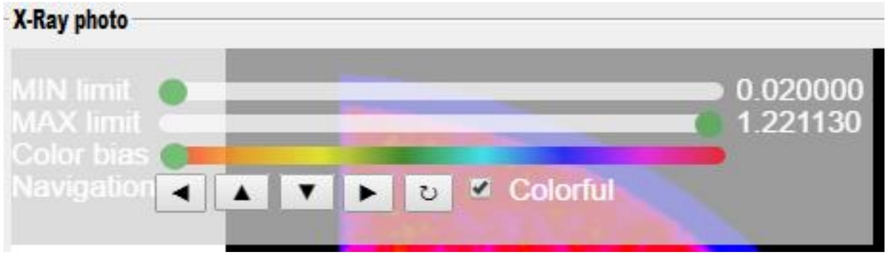
Ray Tracer



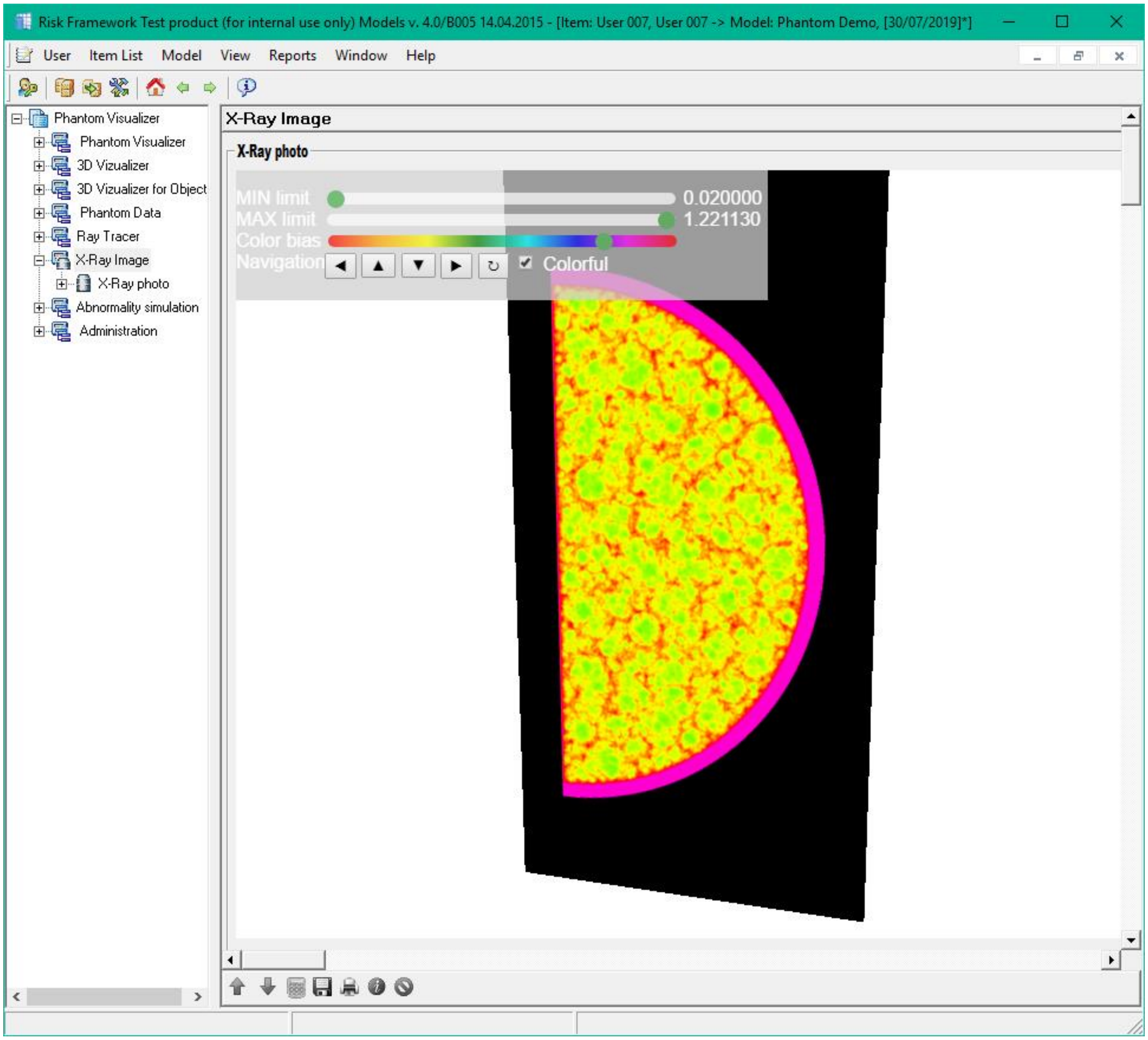
This is the panel for X-ray Photo generation. It contains two buttons:

- Simulates – simulates an x-ray beam and the irradiation of the model. Generates a X-ray image of the model according to the different given parameters that are described in **Irradiation of the phantom**. The result of the simulation if a x-ray photo that is shown in the next tab X-Ray Image.
- Generate image - this button gets the last generated photo and shows it.

There is a check button that removes some of the functionalities when showing the image in the next tab. If it is on the image is gray scaled and it`s minimum and maximum values cannot be changed. When the button if off in the X-ray image panel there is a small block that contains 2 sliders that gives the user the opportunity to change the minimum and maximum value of the picture, which changes some of the images pixels in black. There is also a navigation buttons that moves the picture. The third slider is shown when the check box colorful is checked. It allows the used to change the base color.

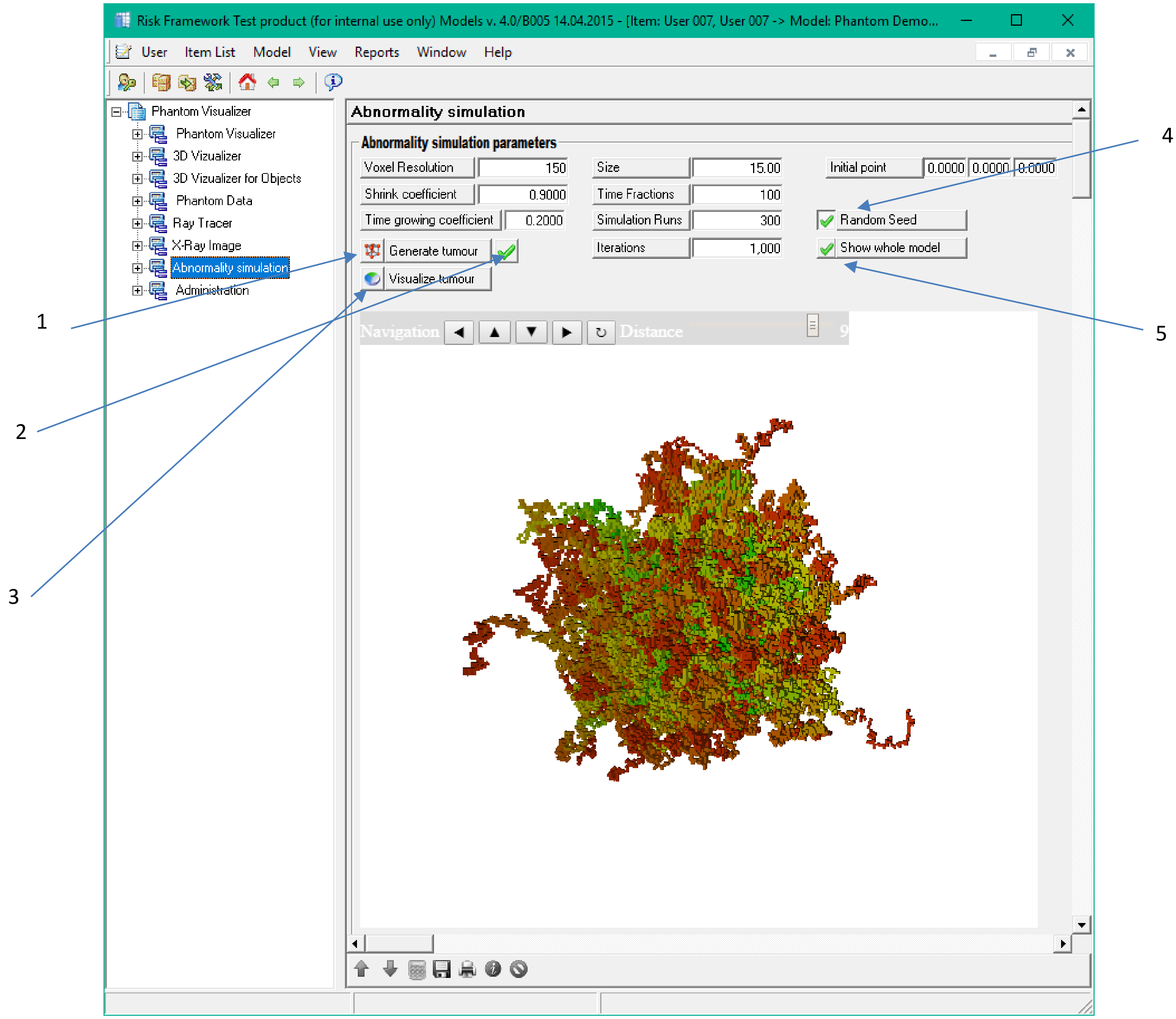


X-ray image



This is the tab where image or images depending of the simulation type, are shown and being changed.

Abnormality Simulation



The parameters from this panel are described in Abnormality. There are also three buttons and two check boxes.

- When button 1 generates an abnormality and then shows it in the frame that contains navigation controls and slider that changes the time phase of the abnormality creation.
- Button 2 applies the last generated tumor on the X-ray photo.
- Button 3 shows the last generated 3D model of tumor in the frame.
- Check box 4 is one of the parameters for the abnormality creation and if it is checked the generated tumor will always be different.

- Check box 5 shows the tumor inside of the model and displays the model with the abnormality in the frame.

12. System Requirements

To use the irradiation library, it is necessary machine to have NVidia Graphic Chip with installed proper driver – Game Ready or studio driver for new model. Nvidia drivers can be downloaded from https://www.nvidia.com/Download/index.aspx

If there is still a crash followed DLLs should be placed - cudart64_90.dll

In addition, visual studio distributives should be installed for runtime - https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads

13. Future developments

One real breast contains not only mass tissue but also mammary gland that does not exist in this model. So the next step is improvement of the model to be closer to the real world. Once the gland is generated into the model, the irradiation library should be modified to detect also glands shapes.

14. Acknowledgement

This product is developed with the irreplaceable help of Dr. Kristina Bliznakova. Model is based on her idea as well as the algorithm for tumor generation. All of the coefficients and part of formulas for the result of X-Ray images are checked by her.