## StarLyX, Monte-Carlo simulation for complex scenes and optical instruments (DRAFT)

Laurent Brunel, PhotonLyX

June 12, 2025

## Contents

## 1 Introduction

Starlyx address to people interested in modelling and measuring light interaction with complex materials. Despite the huge offer for rendering engines, none, to our knowledge, allows flexibility to go beyond synthetic image rendering and simulate optical instruments able to exploit scattering properties of materials. Starlyx allows in one tool to modelise instruments involving diffusive material and to render visual sensation given by volume light scattering inside complex objects. We propose this software, GPL3 licensed, that could be a common tool for people involved in radiative transfer, scattering media studies and visual appearance. Starlyx uses the star-engine library (Meso-Star, Toulouse, France) (star-engine[?]) with itself stands on Embree (Intel)[?]. It is written in C language.

The main points characterizing starlyx are: Monte-Carlo engine dedicated to volume-scattering, spectral simulation, no C coding needed at a start, CPU parallelism, scene and studies described by a single XML file and mesh files (obj), calculation time little dependent on geometry complexity, Monte-Carlo error estimation. Starlyx is initiated by PhotonLyX (Santander,Spain).



Figure 1: starlyx relies on star-engine (meso-star) and Embree (Intel). Illustrative examples: rendering of the Stanford bunny in a foggy cube.

The last release, examples, source code, installation instructions and this doc are available in: https://gitlab.com/photonlyx/starlyx

## 2 Detailed description of the "minimal-cube" example

Run starlyx to create the render of the scattering cube:

```
1 cd examples/minimal-cube
2 starlyx scene.xml
```

```
See the result in cam0.png (figure 2).
```

Let's comment the file scene.xml of this example step by step. The main object of the XML file has the type Scene:



Figure 2: image rendered by the example minimal-cube

```
1 <Scene
2 ALGORITHM="reverse_1"
3 VERBOSE="1"
4 NB_PHOTONS="100"
5 STORE_TRAJECTORIES="0"
6 STORE_ERROR_TRAJECTORIES="0"
7 MAX_PATH_LENGTH="1000"
8 >
```

The attribute ALGORITHM is set to "reverse\_1" to specify that the Monte-Carlo algorithm starts the ray tracing from the sensor (the camera). If ALGORITHM is set to "direct", the ray tracing will start from the sources.

The attribute VERBOSE is set to 1 to get printed output of the calculation process. The attribute STORE\_TRAJECTORIES can be set to 1 to get the photon trajectories in the file paths.obj viewable by any 3D visualisation software like blender of paraview. Use this option only for debugging only since the file "paths.obj" might be very large.

The attribute STORE\_ERROR\_TRAJECTORIES can be set to 1 to get the photon trajectories that have given an error (ray tracing error, infinite loop,...) in the file "paths\_error.obj".

The first son of the Scene is the object of type source named "sourceSun0":

```
1 <source
2 NAME="sourceSun0"
3 RADIANCE="1e7" <!-- W m-2 sr-1 -->
4 TYPE="sun"
5 ANGLE="0.5" <!-- deg -->
6 SPECTRUM="red"
7 >
8 <dir X="0.2" Y="0.3" Z="-1.0"></dir>
9 </source>
```

The attribute TYPE refers to the source type: here the type "sun" which is an infinitely wide source illuminating in a defined direction. This direction is mentioned in the son object of type "dir". The direction does not need to be normalised since it will be automatically done by the software. The attribute ANGLE set the angular diameter of the sun (typically  $0.5^{\circ}$ ). The attribute POWER refers to the source power, in W/m<sup>2</sup> for this type. The attribute "SPECTRUM" refers to the source power for each wavelength in case of a spectral simulation. In this case the simulation is monochromatic and the spectrum refers to the object of type "spectrum" called "red" which has a single wavelength of 650nm with the value of 1 (W/m<sup>2</sup>).

The camera that produce the rendering is described in the object of type "camera" and NAME "cam0"

```
1 <camera
     NAME = " cam0 "
2
     NBPIXELSY = "200"
3
     NBPIXELSX = "200"
 4
     FIELD="30.0" <!-- degrees -->
5
     FILTERS="filterR"
6
7 >
     <lpre><lens Z = "3.0" Y = "-3.0" X = "-3.0">
8
     </lens>
9
     < viewPoint Z = "0.5" Y = "0.5" X = "0.5" >
10
     </viewPoint>
11
12 </camera>
13
14 <spectrum
15 NAME = "filterR"
```

```
16 DATA="650 1"
17 >
18 </spectrum>
```

The attributes NBPIXELSX and NBPIXELSY describe the size the image rendered. The attribute "FIELD" indicates the camera field in degrees. The attribute "FILTERS" refers to the spectral filters used by the different spectral channels of the camera. In this case, we have a monochromatic camera using a monochromatic filter called "filterR". This filter is defined by the object of type "spectrum" named "filterR", placed later in the XML description. The object "cam0" has 2 sons, of type "lens" and viewPoint. The son of type "lens" refers to the position of the camera lens in scene's reference frame. The son of type "viewPoint" indicates the position of the central point targeted by the camera in the scene's reference frame.

The scene contains only one physical object of cubic shape. The surface of this shape is defined by obj file indicated in the attribute "FILE":

```
1 <surface
2 NAME="cube_surface"
3 MATERIALS="dielectric1"
4 FILE="cube.obj">
5 </surface>
```

The material of the surface is indicated in the attribute "MATERIALS" which may contain a list of surface materials describing the behaviour of the photon when it hits this surface. In this case, "MATERIALS" refers to the object of name "dielectric1" which type is "dielectric":

```
<dielectric NAME="dielectric1" ></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></diele
```

As a consequence, the photon will reflect or refract following the Fresnel laws (ignoring polarization). The needed refractive indices are indicated in the volumes descriptions of the XML file (next). For the moment, the other types of surface material availables are "mirror" and "lambert". They are explained later in this paper. Note that the attribute "MATERIALS" might be empty and in this case the photon will have no interaction with the surface.

The unique volume in the scene is delimited by only one surface: the one which name is "cube\_surface". This is indicated in the attribute "SURFACES". The name is in the attribute "NAME", the refractive index in the attribute "N". We will see later that "N" could also refer to a spectrum. The radiative transfer properties of this volume are described in the attribute "MATERIALS" which refers here to another XML object of name "hg2" and type "Henyey-Greenstein".

```
1 <volume
2 NAME="cube_volume"
3 N="1"
4 MATERIALS="hg2"
5 SURFACES="cube_surface">
6 </volume>
```

In this example, the radiative transfer properties are defined by the transport length "LSTAR", the asymetry factor "G" and the absortion length LA. We will see later that other sets of parameters are possible. These parameters may also be spectral.

```
1 <Henyey-Greenstein
2 NAME="hg2"
3 LA="10000000" <!-- mm -->
4 LSTAR="0.1" <!-- mm -->
5 G="0"
6 >
7 </Henyey-Greenstein>
```

The last line must close the XML object of type "Scene".

1 </Scene>

## 3 User guide

The geometry of the scene is described by surface and volumes. The volumes must be entirely closed by one or many surfaces (figure 3).



Figure 3: The surfaces delimit the volumes. Volume A is delimited by the surfaces 10 and 11. Volume B is delimited by surface 2. Volume C0 is delimited by surface 32 and volume C1 by surfaces 31 and 32. Volume D0 is delimited by surfaces 42 and 43. Volume D1 by surfaces 41 and 43.

To each volume is associated a material volume (absorption and Henyey-Greenstein scattering properties). To each surface is associated a material surface.

#### 3.1 Surfaces

A surface in the scene is defined by a geometry file with format OBJ or STL. The name of the file is specified in the attribute FILE. A surface doesn't have to enclose a single volume. The external side of the surface is the side where is the normal. The internal side is the other side. The materials of both sides of the surface are defined in the attribute "MATERIALS". The first material indicated is the one of the external side and the second material is the one of the internal side. If just one material is indicated, it is affected to both sides. When no material is indicated (empty attribute ""), the surface is just used as a volume boundary and has no optical effect.

Note: to check the normal side of the surface use blender to visualise it.

Examples:

```
1
  <surface
     NAME = "cube_surface"
 2
     MATERIALS = "dielectric1"
3
     FILE="cube.obj">
     </surface>
 5
6
  <!-- a bunny with no optical effect on its surface -->
7
     <surface
8
       NAME = "bunny
9
       M A T E R I A L S = ""
       FILE="bunny.stl">
12 </surface>
13
14 <!-- a sphere with a white material on both sides -->
15 < surface
    NAME = "sphere"
16
     MATERIALS = "white1"
17
    FILE="sphere.stl">
18
19 </surface>
2.0
_{21} <!-- a plane with a mirror surface on the side of the normal and
      black surface on the other side -->
22
23 <surface
24
    NAME = "plane"
    MATERIALS="mirror1 black"
25
26
   FILE="plane.stl">
27 </surface>
2.8
```

#### 3.1.1 Surfaces modelized with Openscad

A surface can also be directly designed in the scene description using the openscad software [openscad.org]. Openscad (version 2021.01) must be installed and the PATH environment variable must point to it.

```
1 < ! -- a cube meshed with openscad -->
2 <openscad
      NAME = "cube"
3
4
    CODE="cube();"
    MATERIALS = "lambert1"
5
6 >
  </openscad>
7
_{\odot} <!-- a cone with diameters of 1 and 2 mm , with height of 2mm meshed with openscad
       -->
10 <openscad
1\,1
    NAME = "cyl2"
    CODE="translate([0,0,0.5]) cylinder(d1=1,d2=2,h=2,center=false,$fn=100);"
12
    MATERIALS="dielectric1"
13
14 >
15 </openscad>
```

Note that when the open scad code has some double quotes ("), replace them by single quotes like in this example:

```
1 <!-- a 3D text, meshed with openscad using extrusion-->
2 <openscad
3 NAME="starlyx1"
4 CODE="text1='Starlyx'; linear_extrude(1){text(text1,1);} "
5 MATERIALS="dielectric1" >
6 </openscad>
```

#### 3.2 Volumes

A volume is defined by joining one or several surfaces. When more that one surface are used, the surfaces must hermetically close the volume. The radiative transfer properties of the volume is defined in the attribute "MATERIALS" which may contains the names of several objects defining these properties (see chapter "Volume radiative transfer properties"). The attribute "N" specify the refractive index of the material in the volume.

```
1 <!-- A volume enclosed by the surface cube_surface with radiative

2 transfer property called henyey-greenstein1 -->

3 <volume

4 NAME="cube"

5 N="1.33"

6 MATERIALS="henyey-greenstein1"

7 SURFACES="cube_surface">

8 </volume>
```

#### 3.3 Sources

#### 3.3.1 Sun

A sun at zenith with angular diameter of 0.5 degrees can be coded like that:

```
1 <!-- a source of type sun -->
2 <source
3 NAME="sourceSun0"
4 RADIANCE="1e7" <!-- W m-2 sr-1 -->
5 TYPE="sun"
6 ANGLE="0.5" <!-- degrees -->
7 SPECTRUM="red">
8 <dir X="0" Y="0" Z="-1"></dir>
9 </source>
```

#### 3.3.2 Spot

A spot emitting light of diameter 1 mm, a total angle of emission 10 degrees, at position (0,0,0) and emiting light in the direction of X is coded like that:

```
1 <!-- a source of type spot -->
2 < source
    NAME = "sourceSpot0"
3
    POWER = "1.0" <! -- W -->
4
    TYPE = " spot "
    DIAMETER = "1" < ! -- mm -->
6
    ANGLE="10" <!-- degrees -->
    SPECTRUM = "red">
    <pos X = "0" Y = "0" Z = "0" > </pos>
9
    <dir X="1" Y="0" Z="0"></dir>
10
11 </source>
```

#### 3.3.3 Source geometry defined by any extended surface

Here is he code for a spherical source with geometry is set in the file source\_sphere.obj. The light is emitted in a direction around the local normal. The total emission angle around the normal is defined by the attribute "ANGLE". The light is emitted in the normal direction. Use OBJ format for the mesh file for the moment, not STL. See example "minimal-cube-source-surface" for the case of reverse algorithm. See example "photodiodes-source-sphere" for the case of reverse algorithm.

```
1 <!-- a source of type surface -->
2 <source
3  POWER="1.0"
4  NAME="source0"
5  TYPE="surface"
6  FILE="source_sphere.obj"
7  ANGLE="180"
8  SPECTRUM="red"
9  VOLUME="World" >
10 </source>
```

#### 3.4 Sensors

#### 3.4.1 Camera

A camera bundles many sensors (pixels) and is associated with a rendered image. For the moment, the camera sensor can be used only for the reverse path algorithms. See example "minimal-cube".

- NBPIXELSX,NBPIXELSY: number of pixels of the camera in horizontal and vertical direction.
- FIELD: field of view of the camera in degrees.
- FILTERS: list of the filters used for each pixel. A filter is an xml object of type "spectrum" that defines the spectral response of the filter. A typical color camera will have 3 filters for red, green and blue channels.
- NA: the numerical aperture of the camera: sin of the angle form axis to the aperture edge seen from the focal point.
- FOCAL: the focal length of the camera in mm.
- VOLUME: name of the volume where is the camera. For the moment, Starlyx doesn't detect automatically the volume. Put "World" when the camera is in the external volume.
- lens: the position of the lens centre
- viewPoint: the position of the focused point. This point will be the centre of the depth of field.

```
1 <!-- a camera with 200x200 pixels -->
2 <camera
    NAME = " cam0 "
3
    N B P I X E L S Y = " 200 "
4
    NBPIXELSX = "200"
5
    FIELD = " 30.0"
6
    FILTERS="filterR"
7
    NA="0.2"
8
9
    F O C A L = " 2 "
    VOLUME = "World"
10
11 >
12 <lens Z="3.0" Y="-3.0" X="-3.0"></lens>
13 <viewPoint Z="0.5" Y="0.5" X="0.5"></viewPoint>
14 </camera>
15
16 <spectrum
17 NAME = "filterR"
   DATA = "650 1">
18
19 </spectrum>
```

#### 3.4.2 Sensor

A sensor is a disk in space representing a single detector. It is used to calculate the luminance around one point and around one direction. The acceptance is a cone with a defined angular aperture. The probe can be used only for the reverse path algorithms.

- DIAMETER: diameter of the disk in mm
- ANGLE: angle of the acceptance in degrees (full angle: max 180)
- FILTERS: list of the filters used for each pixel. A filter is an xml object of type "spectrum" that defines the spectral response of the filter. It is possible to put the list of wavelengths in nm (integer values) for monochromatic filters.
- VOLUME: name of the volume where is the probe. For the moment, Starlyx doesn't detect automatically the volume. Put "World" when the probe is in the external volume.

```
1 <!-- a disk sensor at (0,0,10) pointing at (0,0,0) -->
2 <sensor
    NAME = "probe1"
3
    DIAMETER = "1"
4
    A N G L E = " 3 0 "
5
6
    FILTERS="650 530 450"
    VOLUME = "World"
7
8 >
    <pos X="0" Y="0" Z="10" > </pos>
9
   <viewPoint X="0" Y="0" Z="0" ></viewPoint>
10
11 </sensor>
12
```

#### 3.4.3 Sensor defined by any surface

A sensor with a different shape than a disk can be defined by precising the attribute FILE with the name of the OBJ format mesh. Note that the sensor is sensitive on the surface side where is the normal.

- ANGLE: angle of the acceptance around the normal in degrees (full angle: max 180)
- FILTERS: list of the filters used for each pixel. A filter is an xml object of type "spectrum" that defines the spectral response of the filter. It is possible to put the list of wavelengths in nm (integer values) for monochromatic filters.
- VOLUME: name of the volume where is the probe. For the moment, Starlyx doesn't detect automatically the volume. Put "World" when the probe is in the external volume.

```
1 <!-- a sensor defined by the surface defined by the mesh surface1.obj -->
2 <sensor
3 NAME="probe1"
4 ANGLE="30"
5 FILTERS="650 530 450"
6 VOLUME="World"
7 FILE="surface1.obj"
8 >
9 </sensor>
10
```

#### 3.5 Volume radiative transfer properties

A volume has a specific refraction index N and may contain various materials. There are 3 ways to describe these properties:

- (k,g,ka): scattering coefficient (mm-1), asymmetry coefficient, absorption coefficient (mm-1)
- (l\*,g,la): transport length (mm), asymmetry coefficient, absorption length (mm)
- (d,nr,ni,phi): micro-spheres diameter (mm), refractive index real part, refractive index imaginary part, volume concentration of particles.

The triplet of parameters (k,g,ka) is the one finally used by the Monte-Carlo algorithm. The other parameters options are converted.  $l^*$  is defined by the formula  $l^*=1/k/(1-g)$ . The set of parameters (d,nr,ni,phi) is converted to the triplet (k,g,ka) using the Mie theory. Here is an example of volume cumulating the 3 types of materials:

```
1 <!-- a volume with 3 materials -->
2 <volume
 3
    NAME = "cube"
     N = " 1 "
 4
    MATERIALS="hg1 hg2 mie1 "
5
6 >
7
  </volume >
9 <!-- Henyey-Greenstein RT property defined by KS KA and G -->
10 <Henyey-Greenstein
    NAME = "hg1"
    KS = "1"
12
    KA = "0.1"
13
     G = " 0 "
14
15 >
16 </Henyey-Greenstein >
17
_{18} <!-- Henyey-Greenstein RT property defined by LSTAR LA and G _{--}\!\!\!>
19 <Henyey - Greenstein
   NAME = "hg2"
20
    LSTAR = "1"
21
22
    LA="0.1"
    G = " 0 "
23
24 >
25 </Henyey-Greenstein >
2.6
27 <!-- Suspension of 1nm diameter spheres with 1% volume fraction
28 with phase function calculated by Mie theory
29 -->
30 <Mie
    NAME = "mie1"
31
    D_UM = "0.001"
32
    NR = "2.54"
33
    NI="0.001"
^{34}
    PHI="0.01"
35
36 >
37 </Mie >
```

Let's take the example of a suspension with 2 types of particles modelised by a suspension of 5% in volume of particles of 1  $\mu$ m size (refractive index 1.4) and 2% in volume of size 100 nm (refractive index 1.5).

```
1 <volume
    NAME = "cube"
2
    N = "1"
3
    MATERIALS="type1 type2">
4
5 </volume >
7 <!-- Suspension of 1um diameter spheres with 5% volume fraction
8 with phase function calculated by Mie theory
9 -->
10 <Mie
    NAME = "type1"
11
    D_UM = "1"
12
    NR = "1.4"
1.3
   N I = " 0 "
14
    PHI="0.05" >
15
16 </Mie >
17
18 <!-- Suspension of 0.1um diameter spheres with 2% volume fraction
19 with phase function calculated by Mie theory
20 -->
21 <Mie
22
    NAME = "type2"
    D_UM="0.1"
2.3
    NR = "1.5"
24
    N I = " 0 "
25
    PHI="0.02">
2.6
27 </Mie >
```

#### 3.5.1 Diffusive medium modelised by Henyey-Greenstein using k, ka and g

A volume material can be defined by its scattering (k and g) and absorption (ka) properties. The phase function is defined by Henyey-Greenstein formula [?] and determined by the asymmetry coefficient g. Here is an example of medium with a scattering coefficient k = 1 mm-1, an asymmetry coefficient g=0 and an absorption coefficient ka=0.1 mm-1:

```
1 <!-- Henyey-Greenstein RT property defined by KS KA and G-->
2 <Henyey-Greenstein
3 NAME="hg1"
4 KS="1"
5 KA="0.1"
6 G="0"
7 >
8 </Henyey-Greenstein >
```

All the values can refer to the name of a spectrum object including this parameter as a function of the wavelength  $\lambda$ .

#### 3.5.2 Diffusive medium modelised by Henyey-Greenstein using 1\*, g and la

Another way to define scattering and absorption properties can be using  $l^*$  (transport length in mm) and la (absorption length = 1/ka). Here is an example with  $l^* = 1$  mm, g=0 and la = 0.1 mm.

```
1 <!-- Henyey-Greenstein RT property defined by LSTAR LA and G-->
2 <Henyey-Greenstein
3 NAME="hg1"
4 LSTAR="1"
5 LA="0.1"
6 G="0"
7 >
8 </Henyey-Greenstein >
```

All the values can refer to the name of a spectrum object including this parameter as a function of the wavelength  $\lambda$ .

# 3.5.3 Diffusive medium modelised by Henyey-Greenstein defining a suspension of particles

It is also possible to use Henyey-Greenstein phase function by providing a spherical particle size  $(D\_UM)$  in microns, its complex refractive index (NR + i NI) and the volume fraction (PHI). In this case, the Mie theory [?, ?] is used to calculate the scattering and absorption coefficients.

```
1 <!-- Suspension of 0.1um diameter spheres with 2\backslash% volume fraction
2 with phase function of type Henyey-Greenstein with g calculated by Mie theory
  -->
3
4 <Henyey-Greenstein
    NAME = "hgmie1"
5
    D_UM="1"
6
    NR = "1.5"
    NI="0.001"
8
    PHI="0.01"
9
10 >
11 </Henyey-Greenstein >
```

In case of a polydisperse suspension of spheres, one can define a size distribution adding the attribute SIGMA\_D\_UM. The shape of the distribution is by default with a gaussian shape (normal). For a size distribution with log normal shape, put the attribute DISTRIB\_SHAPE with the value "log normal". Set DISTRIB\_SHAPE to "normal" for a gaussian shape. Here is an example a polydisperse suspension with a mean size of 1um and a standart deviation of 0.1um.

```
<!-- Suspension of polydisperse spheres</pre>
                                                  -->
1
2
  <Henyey-Greenstein NAME="suspension1"
    D_UM="1"
3
    SIGMA_D_UM="0.1"
     DISTRIB_SHAPE="normal"
5
    NR = "1.5"
6
    NI="0"
    PHI="0.001"
8
    K A = " 0 "
9
10 >
11 </Henyey-Greenstein>
```

After execution of starlyx, the size distribution can be found in the file "suspension1\_polydispersity.txt". The curve can be seen with gnuplot with the command:

gnuplot -p -e ' plot "suspension1 polydispersity.txt" with lines'.

#### 3.5.4 Spherical particle suspension using the phase function of Mie theory

If the scattering volume is made of micro-metrical spherical particles, the properties can be modelised using the phase function provided by the Mie theory [?, ?] with the sphere of diameter  $D_UM$  and the complex refractive index (real part NR and imaginary part NI) of the sphere's material. Here is an example for a suspension of micro-spheres of Titane-dioxyde of diameter 1um, 1% in volume :

```
1 <!-- Suspension of 1um diameter spheres with 1% volume fraction
2 with phase function calculated by Mie theory
3 -->
4 <Mie
    NAME = "mie1"
5
    D_UM = "1"
6
    NR = "2.54"
    NI="0"
8
    PHI="0.01"
9
    N A N G = " 9 0 "
10
11 >
12 </Mie >
```

The attribute NANG is the number of angles in the PI/2 sector used in Mie calculation.

All the values can be replaced by the name of a spectrum object including this parameter as a function of the wavelength  $\lambda$ .

In this case also one can define a size distribution adding the attribute SIGMA\_D\_UM. The shape of the distribution is by default with a gaussian shape (normal). For a size distribution

with log normal shape, put the attribute DISTRIB\_SHAPE with the value "log normal". Set DISTRIB\_SHAPE to "normal" for a gaussian shape. Here is an example a polydisperse suspension with a mean size of 1um and a standard deviation of 0.1um.

```
_1 <!-- Suspension of polydisperse spheres with mean diameter of 1um , sigma=0.1um
2 with 0.1% volume fraction
3 with phase function calculated by Mie theory
4 -->
5 <Mie NAME="suspension1"
6
    D_UM = "1"
    SIGMA_D_UM = "0.1"
7
    DISTRIB_SHAPE = "normal"
8
9
    NR = "1.5'
    N I = " 0 "
1.0
    PHI="0.001"
    K A = " 0 "
12
13 >
14 </Mie>
```

After execution of starlyx, the size distribution can be found in the file "suspension1\_polydispersity.txt". The curve can be seen with gnuplot with the command:

gnuplot -p -e 'plot "suspension1 polydispersity.txt" with lines'.

#### 3.5.5 Diffusive medium modelised by a user-defined phase function

A volume material can be defined by its scattering (ks) and absorption (ka) properties. The phase function is defined by the user with a sampling for some angles (in degrees from 0 to 180). The values of the phase function for the missing angles are estimated by using linear interpolation. Here is an example of medium with a scattering coefficient k = 1 mm-1, a phase function defined by the curve pf1 and an absorption coefficient ka= 0.1 mm-1 :

```
1 < !-- A volume material with customized phase function -->
2 <Scattering
    N A M E = " m v 1 "
3
    KS="1" <!-- mm-1 -->
4
    PHASE_FUNCTION = "pf1"
5
6
    KA="0.1" <!-- mm-1 -->
7 >
8 </Scattering >
9
10 <!-- A customized phase function -->
11 <sampled_data
   NAME = "pf1"
12
    DATA = "0 10
                  10 10 20 10 30 0
180 0 "
                                             60 0
                                                     90 0
13
                                                            120
14
    0
        150 0
                   180 0
15 >
16 </sampled_data>
```

The phase function will automatically be resampled with a regular step of 1 degree from 0 to 180 degrees.

#### **3.6** Surface properties:

A surface is described by the xml object of type "surface". The attribute "NAME" defines the xml object name. The attribute "FILE" contains the name of the obj file defining the 3D mesh of the surface. Its attribute "MATERIALS" contains the material describing the light behaviour when intersecting with it. The attribute "MATERIALS" can be empty: the surface won't interact with light. Here is an example of surface which 3D mesh is defined by the file "cube.obj" and which optical behaviour is defined by the xml object of name "dielectric1":

```
1 <!-- A cubic surface with dielectric surface material -->
2 <surface
3 NAME="cube_surface"
4 MATERIALS="dielectric1"
5 FILE="cube.obj">
6 </surface>
7 <dielectric NAME="dielectric1" > </dielectric>
```

The available types of material properties are "lambert", "dielectric" and "mirror".

#### 3.6.1 Lambertian surface: lambert

A lambertian (matt) surface is described by the xml object of name lambert. Its attribute "ALBEDO" is in the [0,1] range. For example, if albedo=0.5, the light ray has 50% probability to be absorbed by the surface when hitting it. The albedo attribute can be replaced by the name of a spectrum object describing the albedo as a function of the wavelength  $\lambda$ .

```
1 <!-- A lambertian surface material -->
2 <lambert NAME="lambert1" ALBED0="1"> </lambert>
```

#### 3.6.2 Dielectric interface: dielectric

An interface between 2 dielectric media is described by the xml object of name "dielectric".

```
1 <!-- A dielectric surface material -->
2 <dielectric NAME="dielectric1" > </dielectric>
```

According to the refractive index of the 2 media, light may refract, reflect using the Fresnel laws or totally reflect is incidence is beyond the Brewster angle. Let's note i1 the incident angle in the medium of refractive index n1 and i2 the exit angle in the medium of refractive index n2. The exit angle i2 follows the Snell-Descartes law:

$$n1sin(i1) = n2sin(i2)$$

The reflexion coefficient R is calculated using the Fresnel law making an average of both polarisations:  $((1, 1, \dots, (1, 1)))^2$ 

$$Rs = \left(\frac{(n1 * \cos(i1) - n2 * \cos(i2))}{(n1 * \cos(i1) + n2 * \cos(i2))}\right)^{2}$$
$$Rp = \left(\frac{(n1 * \cos(i2) - n2 * \cos(i2))}{(n1 * \cos(i2) + n2 * \cos(i1))}\right)^{2}$$

$$R = (Rs + Rp)/2$$

When n1  $\sin(i1) > n2$ , i1 is the greater than the Brewster angle: the reflexion is total, R=1 and i2=i1.

#### 3.6.3 Mirror

A reflecting surface is described by the xml object of name "mirror". The attribute R indicates the reflection coefficient. R attribute can be the name of a spectrum described in another place of the xml file.

```
1 <!-- A mirror surface material -->
2 <mirror NAME="mirror1" R="0.95" > </mirror>
```

#### 3.6.4 Surface emission

To simulate a surface that emits light, just put an object of type "emission" in its attribute "MATERIALS". The unit of the corresponding spectrum is  $W/m^2$ . Note: this type of surface is not yet implemented for the direct algorithm (see section ??).

```
1 <!-- A mirror surface material -->
2 <emission NAME="emission1"</pre>
```

```
3 SPECTRUM="spectrum_emission1" > </emission>
```

#### 3.7 Spectral data

Raw spectral data are configured in the scene with objects of type < spectrum> </ spectrum>. The attribute DATA contains the spectral values in format " $\lambda$ 1 v1  $\lambda$ 2 v2 .....". The unit for wavelengths is nm. Here is an example of RGB like spectrum with twice more power on green than other wavelengths:

```
1 <!-- A spectrum with 3 wavelengths -->
2 <spectrum
3 NAME="white"
4 DATA="450 0.25 530 0.5 650 25"
5 >
6 < <spectrum>
```

6 </spectrum>

For the case of the "direct path" algorithm, the number of wavelength simulated is guided by the camera configuration by the way of its channels defined in the xml attribute "FILTERS". We have 3 typical cases: monochromatic simulation with on single channel defined for example by FILTERS="650", RGB simulation with 3 channels defined by FILTERS = "filterR filterG filterB" (referring to xml object of type "spectrum") or directly by FILTERS = "650 530 450", spectral simulation defined for example by FILTERS="700 680 660 640 620 600 580 560 540 520 500 480 460 440 420 400".

For the case of "direct path", the number of wavelength simulated is guided by the source spectrum configuration.

#### 3.7.1 Spectral camera

A 3 wavelengths camera can be directly specified including the 3 wavelengths (in nm) in the attribute "FILTERS" of the xml object of type "camera".

```
1 <!-- A RGB camera -->
2 <camera
3 NAME="cam0"
4 NBPIXELSY="50"
5 NBPIXELSX="50"
6 GAMMA="1"
7 FIELD="30"
8 FILTERS="650 530 450"
9 >
10 <lens X="1" Y="1" Z="1"></lens>
11 <viewPoint X="0" Y="0" Z="0"></viewPoint> </camera>
```

The filters for each camera channel can be specified addressing the 3 spectrums of each channel, like in the next example (see curves in figure 4):

```
_1 <!-- A RGB camera with customized bandpass filters for the red green and blue
2 channels -->
3 <camera
     NAME = "cam0"
     NBPIXELSY = "200"
5
     NBPIXELSX = "200"
6
     FIELD = " 30 "
     FILTERS="filterR filterG filterB">
8
     <lens X="1" Y="1" Z="1"></lens>
Q
     <viewPoint X="0" Y="0" Z="0"></viewPoint>
10
11 </camera>
12
     <spectrum</pre>
13
       NAME="filterR"
14
       DATA="400 0 410 0 420 0 430 0 440 0 450 0 460 0 470 0 480 0 490 0
15
       500 0 510 0 520 0 530 0 540 0 550 0 560 0 570 10 580 30 590 50 600 70 610 80
       620
       90 630 90 640 80 650 70 660 50 670 30 680 10 690 0 700 0"
18
     >
19
     </spectrum>
20
21
     < spectrum</pre>
       NAME = "filterG"
22
       DATA="400 0 410 0 420 0 430 0 440 0 450 0 460 0 470 0 480 0 480 0
23
```

```
490 10 500 30 510 50 520 70 530 80 540 90 550 90 560 81 570 70 580 50 590 30
24
      600
      10 610 0 620 0 630 0 640 0 650 0 660 0 670 0 680 0 690 0"
25
    >
26
    </spectrum>
27
     < spectrum</pre>
2\,8
      NAME = "filterB"
29
      DATA="400 0 410 10 420 30 430 50 440 70 450 80 460 90 470 90 480
30
31
       81 490 70 500 50 510 30 520 10 530 0 540 0 550 0 560 0 570 0 580 0 590 0 600 0
       610 0 620 0 630 0 640 0 650 0 660 0 670 0 680 0 690 0 700 0"
32
33
    >
34 </spectrum>
```

For a 3 channels camera, starlyx will create a color image cam0.ppm compatible with any image viewer. Starlyx will also create the raw format image cam0.raw (format explained in annex).



Figure 4: Example of filters for a RGB camera

#### 3.7.2 Spectral source

The source description must contain the name of a spectral object in the attribute SPECTRUM. In the next example: the source uses the spectrum called "white":

```
1 < ! - - A white source - - >
2 < source
    NAME = "source0"
3
     POWER = "1.0"
     TYPE = " spot "
5
    DIAMETER = " 0 "
6
     ANGLE="180"
7
     SPECTRUM = "white"
8
9 >
10 <pos X="0" Y="0" Z="0"></pos>
11 </source>
12
13 < spectrum
    NAME = "white"
14
     DATA="450 1 530 1 650 1"
1\,5
16 >
17 </spectrum>
```

For spectral simulation, the camera spectrum can be directly set to the CIE D65 illuminant [?] using the syntax:

#### 3.8 Path Algorithms

Three path algorithms are available: direct path, simple reverse path and In the scene XML description, the choice of algorithm is indicated by the attribute "ALGORITHM" of the XML object "Scene". It is set to "direct" for direct path algorithm and "reverse\_1" or "reverse\_2" for reverse path algorithm.

#### 3.8.1 Direct path

For this algorithm the paths start at the sources. All the surfaces of the scene are considered as sensors. When the path is absorbed by a surface (albedo <1) the Monte-Carlo weight is incremented by 1. The surfaces with no material, dielectric or having an albedo of 1 will always get a resulting weight of 0. The results are in the file "sensors.csv" as a list of the weights and errors for each surface.



Figure 5: Example of simulation using direct path (see example "photodiodes")

#### 3.8.2 Reverse path

Like the previous one, to use this algorithm, the scene must contain a camera or some sensors ("probes"). The paths start from the probes inside the acceptance cone or at the camera lens aperture in the direction given by each pixel. The available source type for this algorithm is "spot" or "surface". To accelerate Monte-Carlo convergence for these unextended sources, at each photon scattering inside the volumes, the Monte-Carlo weight is incremented by the source contribution along the direct path from the source to the scattering point. Note that for the moment this algorithm assumes that the path from the scattering point to the source is not deviated, which means that the refractive index of the volume is set to 1. If the index is higher than 1 the rendered image gives a realistic result but not quantitatively exact. The output images files are in png format and raw binary format (see annexes for the detail of this format).



Figure 6: Example of rendering using reverse path

## 4 Examples

Here are described the examples that can be found in the directory "examples" of the repository.

### 4.1 Example "minimal-cube"

The "Hello world" example of Starlyx. This example illustrates a monochromatic render of a diffusive cube illuminated by the sun. Reverse algorithm.



Figure 7: output of the example minimal-cube

```
FILTERS="filterR"
MAX_RADIANCE="-1"
 <?xml version="1.0"?>
<Scene
NB_THREADS="1"
                                                                                                                                                                                                                                                    VOLUME="World"
    ALGORITHM="reverse"
                                                                                                                                                                                                                                                >
    VERBOSE="1"
                                                                                                                                                                                                                                                    <lens X="-3.0" Y="-3.0" Z="3.0"></lens>
   NB_PHOTONS="100"
                                                                                                                                                                                                                                                    <viewPoint X="0.5" Y="0.5" Z="0.55"></viewPoint>
  STORE_TRAJECTORIES="O"
STORE_ERROR_TRAJECTORIES="O"
MAX_NB_EVENTS="1000"
                                                                                                                                                                                                                                                 </camera>
                                                                                                                                                                                                                                                <spectrum
NAME="filterR"</pre>
                                                                                                                                                                                                                                                    DATA="650 1"
 >
<source
NAME="sourceSun0"
TYPE="sun"
ANGLE="0.26"</pre>
                                                                                                                                                                                                                                                    </spectrum>
                                                                                                                                                                                                                                               <surface
NAME="cube_surface"
MATERIALS="dielectric1"</pre>
   RADIANCE="1"
SPECTRUM="red"
                                                                                                                                                                                                                                                   FILE="cube.obj"
    VOLUME="World"
>
                                                                                                                                                                                                                                                 </surface>
   <dir X="0.2" Y="0.3" Z="-1.0"></dir>
                                                                                                                                                                                                                                                 <dielectric NAME="dielectric1" > </dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></diel
                                                                                                                                                                                                                                                <volume
NAME="cube_volume"
</source>
<spectrum
   NAME="red"
DATA="650 1"
                                                                                                                                                                                                                                                    N="1"
                                                                                                                                                                                                                                                    MATERIALS="hg2"
                                                                                                                                                                                                                                                    SURFACES="cube_surface"
 </spectrum>
                                                                                                                                                                                                                                                >
<camera
NAME="camO"
FOCAL="1"
                                                                                                                                                                                                                                                 </volume>
                                                                                                                                                                                                                                               <Henyey-Greenstein NAME="hg2"
LA="10001"</pre>
    N A="O"
                                                                                                                                                                                                                                                    LST AR="0.1"
    NBPIXELSX="100"
                                                                                                                                                                                                                                                    G="0"
   NBPIXELSY="100"
                                                                                                                                                                                                                                                >
   GAMMA="1"
FIELD="30.0"
                                                                                                                                                                                                                                                 </Henyey-Greenstein>
                                                                                                                                                                                                                                                </Scene>
```

### 4.2 Example "minimal-cube-rgb"

This example illustrates a RGB color render. The source (sun) spectrum is white: same power for red, green and blue. Reverse algorithm.



Figure 8: image rendered of the example minimal-cube with 3 wavelengths

<?xml version="1.0"?> <Scene
NB\_THREADS="1"
ALGORITHM="reverse"</pre> VERBOSE="1" NB\_PHOTONS="100" STORE\_TRAJECTORIES="0" STORE\_ERROR\_TRAJECTORIES="O" MAX\_NB\_EVENTS="1000" > <source NAME="sourceSun0" TYPE="sun" SPECTRUM="white" VOLUME="World" ANGLE="0.26" RADIANCE="15000000" <dir X="0.2" Y="0.3" Z="-1.0"></dir> </source> <camera NAME="cam0" VOLUME="World" NBPIXELSY="100" NBPIXELSX="100" G AMM A = " 1 " FIELD="40" FOCAL="1" NA="0" MAX\_RADIANCE="-1" FILTERS="650 530 450" <lens X="3.0" Y="2.0" Z="2.0"></lens> <viewPoint X="0.5" Y="0.5" Z="0.5" ></viewPoint> </camera> <spectrum

```
NAME="white"
      DATA="450 1 530 1 650 1"
>
</spectrum>
 <volume
   NAME="cube_volume"
      N="1"
      MATERIALS="hg1"
      SURFACES="cube_surface"
>
</volume>
<surface
   NAME="cube_surface"
MATERIALS="dielectric1"
   FILE="cube.obj"
 </surface>
 <dielectric NAME="dielectric1" > </dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></diel
<Henyey-Greenstein
NAME="hg1"</pre>
    LA="la"
LSTAR="lstar"
      G="0"
>
 </Henyey-Greenstein>
 <spectrum
   NAME="lstar"
DATA="450 0.5 530 0.5 650 1"
</spectrum>
 <spectrum
   NAME="la"
DATA=" 450 5 530 2.5 650 2.5 "
>
</spectrum>
</Scene>
```

#### 4.3 Example "minimal-cube-spectral"

This example illustrates a render with multi-spectral (7 wavelengths) material properties. The source (a sun) spectrum is the standard D65 spectrum. Reverse algorithm.



Figure 9: image rendered of the example minimal-cube with 7 wavelengths

```
<?xml version="1.0"?>
                                                                                                                                                               <lens X="3.0" Y="2.0" Z="2.0"></lens>
<Scene
NB_THREADS="1"
ALGORITHM="reverse"
                                                                                                                                                               <viewPoint Z="0.5" Y="0.5" X="0.5" ></viewPoint>
                                                                                                                                                             </camera>
                                                                                                                                                             <volume
  VERBOSE="1"
                                                                                                                                                               NAME="cube_volume"
  NB_PHOTONS="100"
                                                                                                                                                               N="1"
  STORE_TRAJECTORIES="0"
                                                                                                                                                               MATERIALS="hg1"
  STORE_ERROR_TRAJECTORIES="O"
                                                                                                                                                               SURFACES="cube_surface"
 MAX_NB_EVENTS="1000"
                                                                                                                                                             </volume>
>
<source
                                                                                                                                                            <surface
 NAME="sourceSun0"
                                                                                                                                                               NAME="cube_surface"
  RADIANCE = "1.0"
                                                                                                                                                               MATERIALS="dielectric1"
 TYPE="sun"
                                                                                                                                                              FILE="cube.obj">
  SPECTRUM="d65"
                                                                                                                                                             </surface>
  ANGLE="O"
                                                                                                                                                             <dielectric NAME="dielectric1" > </dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></diel
 VOLUME="World"
                                                                                                                                                            <Henyey-Greenstein
NAME="hg1"
<dir Z="-1.0" Y="0.3" X="0.2"></dir>
                                                                                                                                                               LA="la"
                                                                                                                                                               LSTAR="lstar"
</source>
<spectrum
                                                                                                                                                               G = "g"
 NAME="white"
                                                                                                                                                            >
 DATA="400 1 450 1 500 1 550 1 600 1 650 1 700 1 "
                                                                                                                                                            </Henyey-Greenstein>
>
                                                                                                                                                             <spectrum
</spectrum>
                                                                                                                                                               NAME="1star"
<camera
                                                                                                                                                               DATA="400 1 450 1.5 500 2 550 2.5 600 3 650 3.5 700 4 "
 NAME="camO"
VOLUME="World"
                                                                                                                                                            >
                                                                                                                                                            </spectrum>
 NBPIXELSY="100"
                                                                                                                                                            <spectrum
NAME="la"
 NBPIXELSX="100"
  G AMM A = " 1 "
                                                                                                                                                              DATA="400 1.0 450 1.5 500 20 550 25 600 30 650 35 700 40 "
 FIELD="40"
 FOCAL="1"
                                                                                                                                                             </spectrum>
 NA="0"
                                                                                                                                                             <spectrum
 FILTERS2="700 650 600 550 500 450 400 "
                                                                                                                                                              NAME="g
 FILTERS="400 450 500 550 600 650 700 "
                                                                                                                                                              DATA="400 0 450 0 500 0 550 0 600 0 650 0 700 0 "
 FILTERS1="650 530 450"
  MAX_RADIANCE="-1"
                                                                                                                                                             </spectrum>
                                                                                                                                                             </Scene>
```

### 4.4 Example "minimal-cube-source-spot"

This example illustrates a render of a cube illuminated by a spot, using the reverse path algorithm.



Figure 10: image rendered of a cube illuminated by a spot

```
<?xml version="1.0"?>
<Scene
NB_THREADS="1"
 ALGORITHM="reverse"
 VERBOSE="1"
NB_PHOTONS="1000"
STORE_TRAJECTORIES="0"
STORE_ERROR_TRAJECTORIES="0"
MAX_NB_EVENTS="1000"
>
<source
NAME="sourceSpot0"
TYPE="spot"
POWER="1"
RADIANCE="1"
DIAMETER="0.5"
ANGLE="20"
SPECTRUM="red"
VOLUME="World"
>
 <pos X="0.5" Y="0.5" Z="2.0"></pos>
<dir X="0" Y="0" Z="-1.0"></dir>
</source>
<spectrum
NAME="red"
DATA="650 1"
>
</spectrum>
<camera
NAME="camO"
FOCAL="1"
NA="0"
NBPIXELSX="100"
NBPIXELSY="100"
```

```
GAMMA="1"
FIELD="30.0"
       FILTERS="filterR"
       MAX_RADIANCE="-1"
      VOLUME="World" >
<lens X="-3.0" Y="-3.0" Z="3.0"></lens>
<viewPoint X="0.5" Y="0.5" Z="0.55"></viewPoint>
 </camera>
  <spectrum
       NAME="filterR"
     DATA="650 1"
 >
       </spectrum>
 <surface
     NAME="cube_surface"
MATERIALS="dielectric1"
      FILE="cube.obj"
 >
 </surface>
 <dielectric NAME="dielectric1" > </dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></dielectric></diel
  <volume
      NAME="cube_volume"
       N="1"
      MATERIALS="hg2"
SURFACES="cube_surface"
 >
 </volume>
<Henyey-Greenstein NAME="hg2"
LA="10000000"</pre>
       LSTAR="0.1"
       G="0"
 >
 </Henyey-Greenstein>
 </Scene>
```

#### 4.5 Example "photodiodes"

This example illustrates the algorithm called "direct".

A laser starting at (0,0,0) and in the direction of Z illuminates a diffusive cube.

A photodiode called "photodiode1" is placed in front of the laser, at the other side of the cube. Another photodiode, called "photodiode2" is placed at a side of the cube.

The surfaces objects of names "photodiode1" and "photodiode2" simulate these photodiodes. They have the MATERIALS property set as "black" which means a lambertian surface of "ALBEDO" 0. All the photons that hits these surfaces will end their path here.

To apply the "direct" algorithm, the xml object of type "scene" has its attribute "ALGO-RITHM" set to "direct".

Its attribute "NB PHOTONS" indicates that 10000 paths are sorted by the Monte-Carlo.



Figure 11: Scene view of a diffusing cube illuminated by a spot type source and 2 photodiodes. Only the paths from the source to the photodiodes are plotted.

After running starlyx with the command "starlyx scene.xml", check the file "sensors.csv" :

photodiode1\_channels: 650
photodiode1\_power 0.00464
photodiode1\_power\_sigma 0.000214907
photodiode2\_channels: 650
photodiode2\_power 0.00152
photodiode2\_power\_sigma 0.000123195

This indicates that, for a laser power of 1, the power received by the photodiode 1 is 0,0048 +/-0.00069 (sigma) and the power received by photodiode 2 is 0.0009 +/-0.00029 (sigma). The cube surface has no materials so the paths never end on the cube.

<?xml version="1.0"?> DATA="650 1" </viewPoint> <Scene > </spectrum> </sensor> NB\_THREADS="1" <sensor ALGORITHM="direct" NAME="photodiode1" VERBOSE="1" DIAMETER="4" NB PHOTONS="100000" ANGLE ="180" STORE\_TRAJECTORIES="0" FILTERS="650" STORE\_ERROR\_TRAJECTORIES="0" VOLUME="World" MAX\_NB\_EVENTS="1000" <pos X="0" Y="0" Z="19"></pos></pos> <viewPoint X="0" Y="0" Z="0"> <source POWER="1.0" </viewPoint> NAME="laser0" </sensor> KS="1" TYPE="spot" KA="0" <sensor DIAMETER="4" NAME="photodiode2" G="0" ANGLE = "3" DIAMETER="4" SPECTRUM="red" ANGLE =" 180" VOLUME="World" > FILTERS="650" VOLUME="World" <dir X="1" Y="0" Z="0"></dir> > <pos X="12" Y="0" Z="7"></pos> </source> </lambert> <spectrum NAME="red"</pre> <viewPoint X="0" Y="0" Z="7"> </Scene>

</revoluts>
</sensor>
</sensor>
</volume NAME="cube0"
N="1.33"
MATERIALS="hg1"
SURFACES="cube"
> </volume>
<surface NAME="cube"
FILE="cube.obj"
MATERIALS=""
> </surface>
<Henyey-Greenstein NAME="hg1"
KS="1"
KA="0"
G="0"
>
</Henyey-Greenstein>
<lambert NAME="black"
ALBED0="0"
>
</lambert>
</Scene>

#### Example "photodiodes-source-sphere" 4.6

This example illustrates the same scene as the previous subsection (photodiodes) replacing the spot source by a source made with a sphere cap. Direct algorithm.



Figure 12: Scene view of a diffusing cube illuminated by a surface type source and 2 photodiodes. Only the paths from the source to the photodiodes are plotted.

<?xml version="1.0"?> <Scene NB\_THREADS="1" ALGORITHM="direct" VERBOSE="1" NB\_PHOTONS="10000" STORE\_TRAJECTORIES="O" STORE\_ERROR\_TRAJECTORIES="0" MAX\_NB\_EVENTS="1000" <source NAME="source0" POWER="1.0" TYPE="surface" FILE="source\_sphere.obj" ANGLE="1" SPECTRUM="red" VOLUME="World" > </source> <spectrum NAME="red"</pre> DATA="650 1" > </spectrum> <sensor
NAME="photodiode1"
DIAMETER="4"
ANGLE="180"</pre> FILTERS="650" VOLUME="World" 5 

</sensor> <sensor NAME="photodiode2" DIAMETER="4" ANGLE="180" FILTERS="650" VOLUME="World" <pos X="12" Y="0" Z="7"></pos>
<viewPoint X="0" Y="0" Z="7"></viewPoint></viewPoint> </sensor> <volume NAME="cube0"
N="1.33"</pre> MATERIALS="mie1" SURFACES="cube" > </volume> <surface NAME="cube"
FILE="cube.obj"</pre> MATERIALS="" > </surface> <Henyey-Greenstein NAME="mie1"
PHI="0.0003"
KA="0"</pre> D\_UM="2" NR="2.54" NI="0" </Henyey-Greenstein> </Scene>

#### 4.7 Example "sensors"

This example illustrates the algorithm called "reverse path". A cube made with scattering material is illuminated by a spot type light source. Three sensors with acceptance of 4PI steradians are placed inside the cube: one at top, one at middle and one at bottom. The results shows how irradiance decreases when going from top to bottom.



Figure 13: The scene of this example

```
sensor1 channels: 650
sensor1_radiance 758.083
sensor1_radiance_sigma 373.397
sensor2_channels: 650
sensor2_radiance 311.191
sensor2_radiance_sigma 104.549
sensor3_channels: 650
sensor3_radiance 134.879
sensor3_radiance_sigma 78.5249
<?xml version="1.0"?>
                                                                <sensor NAME="sensor2"</pre>
                                                                DIAMETER="O"
<Scene
NB_THREADS="1"
                                                                ANGLE="360"
ALGORITHM="reverse"
                                                                FILTERS="650 "
VERBOSE="1"
                                                                VOLUME="cube_volume" >
                                                                 x="5" Y="5" Z="5" > </pos>
NB_PHOTONS="10"
                                                                 <viewPoint X="1000" Y="5" Z="5" ></viewPoint>
STORE_TRAJECTORIES="1"
STORE_ERROR_TRAJECTORIES="0"
                                                                </sensor>
MAX_NB_EVENTS="1000"
                                                                <!-- sensor placed at bottom inside the cube -->
                                                                <!-- 4PI sr acceptance
                                                                                           -->
<!-- spot at the top of the cube -->
                                                                <sensor NAME="sensor3"
<source NAME="source0"</pre>
                                                                DIAMETER="O"
TYPE="spot"
SPECTRUM="red"
                                                                ANGLE="360"
                                                                FILTERS="650 "
DIAMETER="15
                                                                VOLUME="cube_volume" >
ANGLE = "90"
                                                                 <pos X="5" Y="5" Z="1" > </pos>
POWER="1"
                                                                  <viewPoint X="1000" Y="5" Z="5" ></viewPoint>
VOLUME="World"
                                                                </sensor>
                                                                <spectrum NAME="filterR"
DATA=" 650 1 ">
>
 <pos X="5" Y="5" Z="20"></pos>
<dir X="0" Y="0" Z="-1.0"></dir>
                                                                </spectrum>
                                                                <surface NAME="cube_surface"</pre>
</source>
<!-- monochromatic spectrum for the source -->
                                                                MATERIALS="dielectric1"
<!-- lambda = 650 nm
<spectrum NAME="red"
DATA=" 650 1 ">
                                                                FILE="cube.obj">
                        -->
                                                                </surface>
                                                                <dielectric NAME="dielectric1" >
</spectrum>
                                                                </dielectric>
<!-- sensor placed at top inside the cube -->
                                                                <volume NAME="cube_volume"
<!-- 4PI sr acceptance
                                                                N="1"
                          -->
<sensor NAME="sensor1"</pre>
                                                                MATERIALS="hg2"
DIAMETER="O"
                                                                SURFACES="cube_surface">
ANGLE="60"
                                                                </volume>
                                                                <Henyey-Greenstein NAME="hg2"
LA="1000"
FILTERS="650 "
VOLUME="cube_volume" >
 <pos X="5" Y="5" Z="9" > </pos>
                                                                LSTAR="3"
 <viewPoint X="1000" Y="5" Z="5" ></viewPoint>
                                                                G="0">
</sensor>
                                                                 </Henyey-Greenstein>
<!-- sensor placed at middle inside the cube -->
                                                                </Scene>
<!-- 4PI sr acceptance -->
```

## 4.8 Example of modeling a spectrometer

publication\_foin\_et\_al\_2024: Starlyx simulations done by Guillaume Foin during its Phd thesis at Institut Pascal (Université Clermont Auvergne), 2024. See [?].

## 5 Annexes

## 5.1 Format of raw image file

All data is written in float format (4 bytes), little endian.

Header with 7 floats: w=image width (float) h=image height (float) nbc=number of channels (wavelenghts) (float) xmin (0) (float) ymax (1) (float) ymax (1) (float) The pixels data: w\*h\*nbc floats with pixel values. Read it with 3 for loops: for (i=0;i<w;i++) for (j=0;j<h;j++) for (k=0;k<nbc;k++) The wavelenght values: nbc floats with wavelength values in nm

## References

| [Mie 1908]           | Mie, Gustav (1908). Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen .Annalen der Physik. 330 (3): 377-445.   |
|----------------------|---|
| $[\mathrm{Hen}1947]$ | Henyey, L.G., and Greenstein, J.L. (1941), Ap.J.,93, 70.  |
| [Metropolis 1949]    | $\begin{array}{llllllllllllllllllllllllllllllllllll$  |
| [Chandrasekhar 1960] | S. Chandrasekhar. 1960. Radiative transfer. Dover Publications.   |
| [Bohren 2010]        | Bohren, C. F.; Huffmann, D. R. (2010). Absorption and scattering of light by small particles. New York: Wiley-Interscience. ISBN 978-3-527-40664-7.   |
| [Wald 2014]          | Wald, I., Woop, S., Benthin, C., Johnson, G. S., & Ernst, M. (2014, July).<br>Embree: A kernel framework for efficient cpu ray tracing. ACM Trans.<br>Graph., 33 (4), 143:1 143:8. doi: 10.1145/2601097.2601199   |
| [Piaud]              | Piaud B. , Eymet V., Forest V., Coustet C.,<br>star-engine, gitlab.com/meso-star/star-engine  |
| [Foin2024]           | Foin, G.; Brunel, L.; Cornet, JF.; Dauchet, J.; Gros, F.; Vourc'h, T. Extending the Use of Normal Hemispherical Transmittance Measurements by Modeling 3D Multiple Scattering Radiative Transfer. Journal of Quantitative Spectroscopy and Radiative Transfer 2024, 327, 109124. https://doi.org/10.1016/j.jqsrt.2024.109124. |
| [FreeCAD]            | freecad.org   |