Package 'MCBackscattering'

July 21, 2025

Type Package

Title Monte Carlo Simulation for Surface Backscattering

Version 0.1.1

Author Laszlo Baranyai <lbaranyai@atb-potsdam.de>

Maintainer Laszlo Baranyai <Baranyai.Laszlo@etk.szie.hu>

Description Monte Carlo simulation is a stochastic method computing trajectories of photons in media. Surface backscattering is performing calculations in semi-infinite media and summarizing photon flux leaving the surface. This simulation is modeling the optical measurement of diffuse reflectance using an incident light beam. The semi-infinite media is considered to have flat surface. Media, typically biological tissue, is described by four optical parameters: absorption coefficient, scattering coefficient, anisotropy factor, refractive index. The media is assumed to be homogeneous.
Computational parameters of the simulation include: number of photons, radius of incident light beam, lowest photon energy threshold, intensity profile (halo) radius, spatial resolution of intensity profile.
You can find more information and validation in the Open Access paper.
Laszlo Baranyai (2020) <doi:10.1016/j.mex.2020.100958>.

License LGPL-2.1

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-30 10:50:11 UTC

Contents

`Absorb` *Absorbtion of Single Photon Energy in Media*

Description

This function calculates how photon energy decreases during interaction with media.

Usage

```
Absorb(myObject)
```

Arguments

`myObject`	The mandatory parameter of this function is the MCBS class created by `MCBS` constructor function.

Details

The `Absorb` function, alias `Absorb.MCBS`, calculates the new energy level after interaction event in the media. If photon energy falls below 0.001, a roulette algorithm makes decision when photon has 10% chance to survive with higher energy and 90% chance to loose all energy. Please keep in mind, that photons start with `weight = 1` but the first interaction happens at boundary, where photons enter the media.

Value

The `Absorb` function returns an MCBS object as `class`. Typically the same object is used in argument and result. One parameter of a single photon is updated in this procedure:

`weight`	photon energy. Its initial value is 1 and decrease according to the albedo.

Note

Please note that this function is designed to be used in `Simulation` function. You may use it separately in case you design your own procedure. In such a case, please note that function `Randomize` prepares trajectory data for each photon and iterations are done using move index `myObject$midx`. Function `Absorb` uses random data in roulette algorithm to make decision about photon survival at low energy level.

See Also

MCBS for construction of object with initial input parameters. Simulation for running the simulation with adjusted parameters. Chart for plot of calculated photon flux profile. Export for export of photon flux with corresponding radii. Setup for initial computation of specular reflection and transport albedo. Launch for start position of single photon and initial direction in media. Bounce for interaction with surface and computation of photon flux leaving media. Move for moving single photon forward. Scatter for single photon scattering interaction with media. Randomize for adjustment of random trajectory vectors.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
cfgSimulation <- c(1e6,0.05,1e-9,3,0.012)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Setup(apple)
apple <- Randomize(apple)

# launch with 10° incident angle and move first photon
apple$idx <- 1
apple <- Launch(apple,10)
# initial weight in media
cat("Photon energy level",apple$weight,"\n")
# first interaction event
apple$midx <- 1
apple <- Absorb(apple)
cat("Photon energy level",apple$weight,"\n")
```

Bounce *Bounce Photon Interaction with Surface*

Description

This function is applied when photon trajectory goes out of the media or reaches the surface (z <= 0). The photon flux is summarized upon leaving semi-infinite media through the surface.

Usage

```
Bounce(myObject)
```

Arguments

myObject	The mandatory parameter of this function is the MCBS class created by MCBS constructor function.

Details

The Bounce function, alias Bounce.MCBS, calculates internal reflection if incident angle to surface boundary was higher than critical angle. In all other cases, Fresnel reflection is calculated according to the direction and refractive index of media. Remaining photon weight that leaves the media is collected and summarized by the position. Position is calculated as distance from incident point. This way, radial averaging is performed with rigns of width of the spatial resolution. The internally reflected part of photon energy still follows trajectory until reaches limiting energy or expected trajectory length.

Value

The Bounce function returns an MCBS object as class. Typically the same object is used in argument and result. The function collects photon flux in an array:

heat	vector of photon flux scattered back from media. One element represent a ring of width of the spatial resolution. The center of the ring is the incident point.

Note

Please note thet this is an internal function of Simulation. Do not use it separately, unless you build your own simulation procedure. The function uses the coordinates c(x,y,z) of the current photon and the c(u,v,w) direction of its last move.

See Also

MCBS for construction of object with initial input parameters. Simulation for running the simulation with adjusted parameters. Chart for plot of calculated photon flux profile. Export for export of photon flux with corresponding radii. Setup for initial computation of specular reflection and transport albedo. Launch for start position of single photon and initial direction in media. Move for moving single photon forward. Absorb for absorption of single photon energy in interaction with media. Scatter for single photon scattering interaction with media. Randomize for adjustment of random trajectory vectors.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
cfgSimulation <- c(1e6,0.05,1e-9,3,0.012)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Setup(apple)
apple <- Randomize(apple)

# launch and move first photon
apple$idx <- 1
apple <- Launch(apple)
# first move
apple$midx <- 1
```

```
apple <- Move(apple)
if (apple$z <= 0) {
 apple <- Bounce(apple)
}
```

`Chart` *Plot Backscattering Photon Flux Profile*

Description

This function creates a plot of diffuse reflectance (backscattering) intensity measured on the surface. The photon flux is shown by radius (distance from incident point).

Usage

```
Chart(myObject,isLog=FALSE)
```

Arguments

`myObject`	The mandatory parameter of this function is the MCBS class created by `MCBS` constructor function.
`isLog`	Logical variable. Default value is FALSE. If set to TRUE, photon flux will be shown in logarithmic scale.

Details

The `Chart` function, alias `Chart.MCBS`, plots simulation results. Photon flux is computed by `Simulation` function. The chart shows flux according to the spatial resolution of the simulation. In case you wish to make your customized chart, please use the `Export` function to receive data.

Value

This function does not have any result value.

Note

In case you wish to customize your chart, please use `Export` function to receive data.

See Also

`MCBS` for construction of object with initial input parameters. `Setup` for initial computation of specular reflection and transport albedo. `Simulation` for running the simulation with adjusted parameters. `print` show adjusted parameters and total backscattered reflection. `Export` for export of photon flux with corresponding radii.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
# the 100 low number of photons is only for demonstration
cfgSimulation <- c(100,0.05,1e-9,3,0.1)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Simulation(apple)

# show simulation result
Chart(apple)
# show simulation result with logarithmic scale
Chart(apple,TRUE)
```

`Export` *Export Simulation Result*

Description

This function exports simulation results in table format. It can be used to save results into file or make customized charts.

Usage

```
Export(myObject)
```

Arguments

`myObject`	The mandatory parameter of this function is the MCBS class created by `MCBS` constructor function.

Details

The `Export` function, alias `Export.MCBS`, provides a table with column names. Results of Monte Carlo simulation computed by the `Simulation` function are given as `data.frame`. The radii are automatically calculated according to the spatial resolution.

Value

Results are provided as `data.frame` in table format. Two columns represent radii and photon flux.

`Radius`	radius of concentric rings, where photon flux is calculated. Starts with incident point at radius of zero (r=0). The unit is cm.
`Flux`	calculated photon flux, 1/cm^2

See Also

MCBS for construction of object with initial input parameters. Setup for initial computation of specular reflection and transport albedo. Simulation for running the simulation with adjusted parameters. print show adjusted parameters and total backscattered reflection. Chart for plot of calculated photon flux profile.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
# the 100 low number of photons is only for demonstration
cfgSimulation <- c(100,0.05,1e-9,3,0.1)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Simulation(apple)

# show photon flux near incident point
tbl <- Export(apple)
tbl[1:3,]

# save results into file
write.table(Export(apple),tempfile("apple",fileext=".dat"))
```

Launch *Launch Photons into Semi-infinite Media*

Description

This function starts single photon into the semi-infinite media from surface (z = 0). The 3D launch direction c(u,v,w) points into the media, according to the incident angle (relative to normal) and refractive index. If incident angle was zero, the incidence is normal (perpendicular to surface). If incidence angle was not zero, elliptical distortion of circular light beam is also considered.

Usage

```
Launch(myObject,iAngle=0)
```

Arguments

myObject	The mandatory parameter of this function is the MCBS class created by MCBS constructor function.
iAngle	This optional parameter means the incident angle relative to normal. The default value is 0 (the incidence is perpendicular to surface).

Details

The Launch function, alias Launch.MCBS, calculates the start position of the single photon on the surface. The coordinates c(x,y,z) are computed as z = 0 and c(x,y) represent random position in circular light beam. The values z > 0 are inside the media. Beam intensity distribution is considered to be flat, what means that positions have equal probability to occur. Starting polar coordinates are adjusted randomly using uniform distribution by function Setup. If incident angle was not zero, elliptical distortion is corrected along the y-axis.

Value

The Launch function returns an MCBS object as class. Typically the same object is used in argument and result. Position and starting 3D vector are calculated:

x	position along x-axis, calculated from random polar coordinates.
y	position along y-axis, calculated from random polar coordinates. If incident angle was not zero, this value is distorted.
z	initial depth, z=0 on the surface.
u	launch vector coordinate along x-axis.
v	launch vector coordinate along y-axis. If incident angle was not zero, this value is calculated according to the incident angle and refractive index.
w	launch vector coordinate along z-axis. If incident angle was not zero, this value is calculated according to the incident angle and refractive index.

Note

Theoretical models assume the incident angle is zero. During measurement, camera and lightsource cannot occupy the same direction. According to the Fresnel reflection equations, incident angle below 20° is recommended. The non-zero incident angle requires correction in computation by means of refraction of the initial direction inside media and elliptical distortion of circular light beam on the surface. Please note that this function is not intended to use independently. It calculates start values for photon of index myObject$idx. If you want to use it to build your custom simulation, do not forget to assign index, such as myObject$idx <- 1. Index number must be between 1 and myObject$MAXLEN.

References

Jacques, S.L. (1998) Light Distributions from Point, Line and Plane Sources for Photochemical Reactions and Fluorescence in Turbid Biological Tissues. *Photochemistry and Photobiology*, **67(1)**: 23—32. doi: 10.1111/j.17511097.1998.tb05161.x.

Wang, L., Jacques, S.L., Zheng, L. (1997) CONV-convolution for responses to a finite diameter photon beam incident on multi-layered tissues. *Computer Methods and Programs in Biomedicine*, **54**: 141—150. doi: 10.1016/S01692607(97)000217.

See Also

MCBS for construction of object with initial input parameters. Simulation for running the simulation with adjusted parameters. Chart for plot of calculated photon flux profile. Export for export

of photon flux with corresponding radii. Setup for initial computation of specular reflection and transport albedo. Bounce for interaction with surface and computation of photon flux leaving media. Move for moving single photon forward. Absorb for absorption of single photon energy in interaction with media. Scatter for single photon scattering interaction with media. Randomize for adjustment of random trajectory vectors.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
cfgSimulation <- c(1e6,0.05,1e-9,3,0.012)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Setup(apple)

# calculate first photon
apple$idx <- 1
apple <- Launch(apple)
# see vector coordinates
cat("Start position",apple$x,":",apple$y,"\n")
cat("Direction",apple$u,":",apple$v,":",apple$w,"\n")

# calculate first photon with 10° incident angle
apple$idx <- 1
apple <- Launch(apple,10)
# see vector coordinates
cat("Start position",apple$x,":",apple$y,"\n")
cat("Direction",apple$u,":",apple$v,":",apple$w,"\n")
```

MCBS *Monte Carlo Backscattering class*

Description

This function is the constructor of Monte Carlo Backscattering (MCBS) class. Diffuse reflection, also called as backscattering, is the phenomena when light enters into the media, such as biological tissue, and after traveling some distance it leaves the media or tissue through the surface.

Usage

```
MCBS(myTissue,myLight)
```

Arguments

myTissue	collection of optical parameters of the semi-infinite medium. It is a list of numbers: c(mua,mus,g,n) as absorption coefficient (1/cm), scattering coefficient (1/cm), anisortopy factor, refractive index.

`myLight` collection of computational parameters. It is a list of numbers: `c(photons,beamradius,limitingenerg` as number of photons used in calculation, radius of incident light beam (cm), photon energy threshold to stop tracing if energy decreased too much, radius of surface area where photon flux is collected (cm), spatial resolution (cm/pixel). Due to the simulation of optical measurement, spatial resolution is expressed on the basis of pixel (px).

Details

The `MCBS` function creates and initializes a class using the given parameters. The two arguments are separated according to their role. The first one describes media in which photons travel. The second defines simulation parameters. See arguments for details.

The optical parameters describing observed media typically has the relationship of `mua << mus`, which means absorption coefficient is much smaller that scattering coefficient. Many research papers publish the reduced scattering coefficient `rmus = mus*(1-g)`. The general problem of comparison and validation is that `list(mus=2,g=0)` results the same reduced scattering coefficient like `list(mus=20,g=0.9)`. The value of g shows the most likely direction of light after interaction event in media: g = 1 means continue forward, g = 0 means there is no expected direction and media is isotropic, g = -1 means reflected backward. In biological tissues, anisotropy factor (g) is reported to be in the range of 0.8 - 0.99.

Since argument `myLight` is reflecting measurement conditions and hardware setup, it can be considered constant for one system. Please note that quality of results depends on the number of photons. The more is used the better quality you get. The minimum suggested number is 100,000 photons. If you calculate the amount of photons with sensor or camera integration time, this number will increase above billion. The argument `myTissue` includes optical parameters of the sample, this is expected to change during experiment such as storage, grading, etc.

Value

The `MCBS` function returns an object of `class`. Package functions are used to access their parameters for reporting, adjustment and computation. Direct manipulation of object parameters is highly not recommended. Please use the provided functions to reach what is necessary or use them read-only. An object of class `"MCBS"` is a list containing at least the following components:

`mu_a` absorption coeffieint, 1/cm

`mu_s` scattering coefficient, 1/cm

`g` anisotropy factor

`n` relative refractive index of media (the other media is assumed to be air)

`albedo` transport albedo, the descrease of photon energy after one absorption event in media. The value is initialized by `Setup` function.

`cangle` cosine of critical angle. The value is initialized by `Setup` function.

`rs` specular reflection. The value is calculated on the basis of refractive index by `Setup` function.

Other parameters include containers for calculation. Please do not use them directly. See examples how to extract data.

Author(s)

The object design was inspired by the C code "Small Monte Carlo" written by Scott Prahl (http://omlc.org).

References

Jacques, S.L. (1998) Light Distributions from Point, Line and Plane Sources for Photochemical Reactions and Fluorescence in Turbid Biological Tissues. *Photochemistry and Photobiology*, **67(1)**: 23-–32. doi: 10.1111/j.17511097.1998.tb05161.x.

Wang, L., Jacques, S.L., Zheng, L. (1995) MCML – Monte Carlo modeling of light transport in multi-layered tissues. *Computer Methods and Programs in Biomedicine*, **47**: 131-–146. doi: 10.1016/01692607(95)01640F.

Wang, L., Jacques, S.L., Zheng, L. (1997) CONV-convolution for responses to a finite diameter photon beam incident on multi-layered tissues. *Computer Methods and Programs in Biomedicine*, **54**: 141—150. doi: 10.1016/S01692607(97)000217.

Zołek, N.S., Liebert, A., Maniewski, R. (2006) Optimization of the Monte Carlo code for modeling of photon migration in tissue. *Computer Methods and Programs in Biomedicine*, **84**: 50—57. doi: 10.1016/j.cmpb.2006.07.007.

Baranyai, L., Zude, M. (2009) Analysis of laser light propagation in kiwifruit using backscattering imaging and Monte Carlo simulation. *Computers and Electronics in Agriculture*, **69**: 33-–39. doi: 10.1016/j.compag.2009.06.011.

Baranyai, L. (2020) Laser induced diffuse reflectance imaging – Monte Carlo simulation of backscattering measured on the surface. *MethodsX*, **7**: 100958. doi: 10.1016/j.mex.2020.100958.

See Also

`Setup` for initial computation of specular reflection and transport albedo. `Randomize` for adjustment of random trajectory vectors. `Simulation` for running the simulation with adjusted parameters. `print` show adjusted parameters and total backscattered reflection. `Chart` for plot of calculated photon flux profile. `Export` for export of photon flux with corresponding radii.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

cfgMedia <- c(
0.63,  # absorption, 670 nm
30,    # reduced scattering, 670 nm
0,     # isotropic tissue assumed
1.4)   # refractive index

cfgSimulation <- c(
1e6,   # 1 million photons
0.05,  # 1 mm diameter (0.05 cm radius) laser light beam
1e-9,  # limiting energy level
3,     # 3 cm radius is computed
0.012) # 0.012 cm/pixel resolution is assumed
```

```
apple <- MCBS(cfgMedia,cfgSimulation)

cfgMedia <- c(
0.63,  # absorption, 670 nm
300,   # scattering, 670 nm
0.9,   # anisotropic tissue with forward scattering is assumed
1.4)   # refractive index
apple <- MCBS(cfgMedia,cfgSimulation)
```

`Move` *Moves Single Photon in Media*

Description

This function calculates the move of single photon in media.

Usage

```
Move(myObject)
```

Arguments

`myObject`	The mandatory parameter of this function is the MCBS class created by `MCBS` constructor function.

Details

The `Move` function, alias `Move.MCBS`, calculates updated position coordinates `c(x,y,z)` according to the motion vector `c(u,v,w)` and the travel length until next interaction event. The movement of single photon is calculated in 3 dimensional space. The length of the straight section a photon can travel until the next interaction event is estimated as `-log( rnd )` where `rnd` is a uniformly distributed random variable from range 0-1.

Value

The `Move` function returns an MCBS object as `class`. Typically the same object is used in argument and result. As a result, the 3D position of the photon is updated:

`x`	position along x-axis
`y`	position along y-axis
`z`	position along z-axis, the depth

Note

Please note that this function is designed to be used in `Simulation` function. You may use it separately in case you design your own procedure. In such a case, please note that function `Randomize` prepares trajectory data for each photon and steps are done using move index `myObject$midx`.

See Also

MCBS for construction of object with initial input parameters. Simulation for running the simulation with adjusted parameters. Chart for plot of calculated photon flux profile. Export for export of photon flux with corresponding radii. Setup for initial computation of specular reflection and transport albedo. Launch for start position of single photon and initial direction in media. Bounce for interaction with surface and computation of photon flux leaving media. Absorb for absorption of single photon energy in interaction with media. Scatter for single photon scattering interaction with media. Randomize for adjustment of random trajectory vectors.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
cfgSimulation <- c(1e6,0.05,1e-9,3,0.012)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Setup(apple)
apple <- Randomize(apple)

# launch with 10° incident angle and move first photon
apple$idx <- 1
apple <- Launch(apple,10)
# first move
apple$midx <- 1
cat("Position",apple$x,":",apple$y,":",apple$z,"\n")
apple <- Move(apple)
cat("Position",apple$x,":",apple$y,":",apple$z,"\n")
# second move
apple$midx <- 2
apple <- Move(apple)
cat("Position",apple$x,":",apple$y,":",apple$z,"\n")
```

---

print *Print text report on console*

Description

This function prints text report about simulation parameters and calculated coeeficients (albedo, critical angle) and total backscattered reflection.

Usage

```
print(myObject)
```

Arguments

myObject	The mandatory parameter of this function is the MCBS class created by MCBS constructor function.

Details

The print function, alias print.MCBS, provides text description of simulation. Results of Monte Carlo simulation computed by the Simulation function are given as data.frame. Text report includes: absorption coefficient, scattering coefficient, anisotropy factor, refractive index, transport albedo, specular reflection, critical angle, backscattered total reflection.

Value

Results of simulation are provided as data.frame in table format. Print function lists adjusted parameters on console, does not return any value.

See Also

MCBS for construction of object with initial input parameters. Setup for initial computation of specular reflection and transport albedo. Simulation for running the simulation with adjusted parameters. Chart for plot of calculated photon flux profile. Export for export of photon flux with corresponding radii.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
# the 100 low number of photons is only for demonstration
cfgSimulation <- c(100,0.05,1e-9,3,0.1)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Simulation(apple)

# show simulation report
print(apple)
```

Randomize *Random Numbers Selection for Single Photon Trajectory*

Description

This function prepares vectors of random numbers to accelerate trajectory computation. Random numbers are selected from uniform distribution.

Usage

```
Randomize(myObject)
```

Arguments

myObject	The mandatory parameter of this function is the MCBS class created by MCBS constructor function.

Details

The Randomize function, alias Randomize.MCBS, puts random numbers into the container vectors created with the object. During trajectory computation, each interaction event occurs randomly. Prepared container vectors accelerate computation, since values for each event are accessible with myObject$midx move index. Vector length myObject$MAXLEN is adjusted by Setup function based on transport albedo and limiting energy.

Value

The Randomize function returns an MCBS object as class. Typically the same object is used in argument and result. There are five containers of random numbers. Values are in the range of 0 to +1 unless indicated in description.

rmv	length of linear section between interaction events. Used by Move function.
rabs	decision values for roulette algorithm. Used by Absorb function.
rx1	new scattering direction in the range of -1 to +1. Used by Scatter function.
rx1	new scattering direction in the range of -1 to +1. Used by Scatter function.
rmu	variable for Heyney-Greenstein phase function. Used by Scatter function.

Note

Please note that this function is designed to be used in Simulation function. You may use it separately in case you design your own procedure. In such a case, please note that function Randomize prepares random values for one photon and has to be run always before calculating single photon trajectory.

See Also

MCBS for construction of object with initial input parameters. Simulation for running the simulation with adjusted parameters. Chart for plot of calculated photon flux profile. Export for export of photon flux with corresponding radii. Setup for initial computation of specular reflection and transport albedo. Move for moving single photon forward. Absorb for absorption of single photon energy in interaction with media. Scatter for single photon scattering interaction with media.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
cfgSimulation <- c(1e6,0.05,1e-9,3,0.012)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Setup(apple)
```

```
# first five random values of move length
apple <- Randomize(apple)
apple$rmv[1:5]
```

Scatter *Scattering of Single Photon in Media*

Description

Scattering is the interaction event in the media when photon hits structural elements and changes direction as a result of collision.

Usage

```
Scatter(myObject)
```

Arguments

myObject	The mandatory parameter of this function is the MCBS class created by MCBS constructor function.

Details

The Scatter function, alias Scatter.MCBS, calculates the new direction of photon travel. In isotropic media, any new 3D direction can occur with equal probability. In anisotropic media, the new direction is decided according to the Heyney-Greenstein phase function. The isotropic and anisotropic behavior of media is shown by anisotropy factor (g). The anisotropy factor is zero (g = 0) for isotropic media, g = -1 means backward reflection (backscattering) is the most likely, while g = +1 means the most likely result is that photon continues forward (forward scattering). Biological tissues were reported to have high positive anisotropy factor g > 0.8.

Value

The Scatter function returns an MCBS object as class. Typically the same object is used in argument and result. As a result of Scatter function, new direction is given by c(u,v,w). These values are used in function Move to calculate next position.

u	3D vector coordinate along x-axis
v	3D vector coordinate along y-axis
w	3D vector coordinate along z-axis

Note

Please note that this function is designed to be used in Simulation function. You may use it separately in case you design your own procedure. In such a case, please note that function Randomize prepares trajectory data for each photon and iterations are done using move index myObject$midx. Results of Scatter function are used in Move function.

References

Henyey, L.G., Greenstein, J.L. (1941) Diffuse Radiation in the Galaxy. *Astrophysical Journal*, **93**, 70–83. doi: 10.1086/144246.

Jacques, S.L. (1998) Light Distributions from Point, Line and Plane Sources for Photochemical Reactions and Fluorescence in Turbid Biological Tissues. *Photochemistry and Photobiology*, **67(1)**: 23-–32. doi: 10.1111/j.17511097.1998.tb05161.x.

See Also

`MCBS` for construction of object with initial input parameters. `Simulation` for running the simulation with adjusted parameters. `Chart` for plot of calculated photon flux profile. `Export` for export of photon flux with corresponding radii. `Setup` for initial computation of specular reflection and transport albedo. `Randomize` for adjustment of random trajectory vectors. `Launch` for start position of single photon and initial direction in media. `Bounce` for interaction with surface and computation of photon flux leaving media. `Move` for moving single photon forward. `Absorb` for absorption of single photon energy in interaction with media.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
cfgSimulation <- c(1e6,0.05,1e-9,3,0.012)
apple <- MCBS(cfgMedia,cfgSimulation)
apple <- Setup(apple)
apple <- Randomize(apple)

# launch with 10° incident angle and move first photon
apple$idx <- 1
apple <- Launch(apple)

# first interaction event
cat("Direction",apple$u,":",apple$v,":",apple$w,"\n")
apple$midx <- 1
apple <- Scatter(apple)
cat("Direction",apple$u,":",apple$v,":",apple$w,"\n")
```

`Setup` *Initial Configuration of MCBS Class*

Description

This function performs the basic configuration of Monte Carlo Backscattering (MCBS) class. It calculates specular reflection, transport albedo, critical angle and similar parameters.

Usage

```
Setup(myObject)
```

Arguments

myObject	The mandatory parameter of this function is the MCBS class created by MCBS constructor function.

Details

The function Setup, alias Setup.MCBS, performs the initial calculation of essential parameters for computation. Additionally, computes the maximum expected trajectory length and prepares the launch position of all photons with random polar coordinates within circular beam.

Value

The Setup function returns an MCBS object as class. Typically the same object is used in argument and result. The new parameters calculated in Setup are:

albedo	transport albedo
rs	specular reflection
cangle	cosine value of critical angle
MAXLEN	maximum expected length of photon trajectory. This length depends on the albedo and limiting energy.

There are additional containers for random launch position of photons. Launch position is given with polar coordinates (radius, angle).

Note

Please note that this function is designed to be used in Simulation function. You may use it separately in case you design your own procedure.

See Also

MCBS for construction of object with initial input parameters. Simulation for running the simulation with adjusted parameters. Chart for plot of calculated photon flux profile. Export for export of photon flux with corresponding radii. Launch for start position of single photon and initial direction in media. Randomize for adjustment of random trajectory vectors.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862

# create object
cfgMedia <- c(0.63,30,0,1.4)
cfgSimulation <- c(1e6,0.05,1e-9,3,0.012)
apple <- MCBS(cfgMedia,cfgSimulation)
```

```
apple <- Setup(apple)
# transport albedo
apple$albedo
# specular reflection
apple$rs
```

`Simulation` *Runs the Monte Carlo Simulation*

Description

This function performs all actions to run the simulation. It calls other functions that prepare computation and calculate photon trajectory. Please keep in mind, that this technique can be slow depending on the number of photons and computational power of your hardware.

Usage

```
Simulation(myObject,iAngle=0)
```

Arguments

`myObject`	The mandatory parameter of this function is the MCBS class created by `MCBS` constructor function.
`iAngle`	The optional parameter `iAngle` defines the incident angle of light beam. It is measured from normal position, perpendicular to the surface. The default value is zero (iAngle=0). This parameter is passed to `Launch` function.

Details

The function `Simulation`, alias `Simulation.MCBS`, was designed to take every steps such as `Setup` initial conditions and compute each photon trajectory individually. It was written keeping clarity in mind. In case you want to make your own procedure, this function can be used as a template.

The result of simulation is photon flux, which is expressed as relative fluence rate (F/P, 1/cm^2) where F is the fluence rate and P is the power of light source. The fluence rate (F) is computed in concentric rings around the incident point.

Value

The `Simulation` function returns an MCBS object as `class`. Typically the same object is used in argument and result. The main objective of the simulation is to calculate spatial photon flux. The flux is computed with radial averaging relative to the incident point. Each value represent the average flux of the ring with the width of resolution (1 pixel).

`heat`	calculated photon flux, 1/cm^2

Note

Please note that quality of results depends on the number of photons. The more is used the better quality you get. The minimum suggested number is 100,000 photons. If you calculate the amount of photons with sensor or camera integration time, this number will increase above billion. Validation of computations is difficult, because other Monte Carlo programs or the diffusion theory model can be used as reference. The diffusion theory model does not consider many parameters included in Monte Carlo simulation, such as beam radius, incident angle, anisotropy. Please note that models and reports usually use the reduced scattering coefficient `rms = (1-g)*ms` what means that different scattering and anisotropy parameters, like `list(ms=20,g=0)` and `list(ms=200,g=0.9)`, result the same reduced scattering coefficient. Do not access and manipulate the result data vector `myObject$heat` directly. You can receive results by using the `Export` function.

References

Farrell, T.J., Patterson, M.S., Wilson, B. (1992) A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo. *Medical Physics*, **19(4)**: 879–888. doi: 10.1118/1.596777.

Jacques, S.L. (1998) Light Distributions from Point, Line and Plane Sources for Photochemical Reactions and Fluorescence in Turbid Biological Tissues. *Photochemistry and Photobiology*, **67(1)**: 23-–32. doi: 10.1111/j.17511097.1998.tb05161.x.

Wang, L., Jacques, S.L., Zheng, L. (1995) MCML – Monte Carlo modeling of light transport in multi-layered tissues. *Computer Methods and Programs in Biomedicine*, **47**: 131-–146. doi: 10.1016/01692607(95)01640F.

Wang, L., Jacques, S.L., Zheng, L. (1997) CONV-convolution for responses to a finite diameter photon beam incident on multi-layered tissues. *Computer Methods and Programs in Biomedicine*, **54**: 141—150. doi: 10.1016/S01692607(97)000217.

Zołek, N.S., Liebert, A., Maniewski, R. (2006) Optimization of the Monte Carlo code for modeling of photon migration in tissue. *Computer Methods and Programs in Biomedicine*, **84**: 50—57. doi: 10.1016/j.cmpb.2006.07.007.

Baranyai, L., Zude, M. (2009) Analysis of laser light propagation in kiwifruit using backscattering imaging and Monte Carlo simulation. *Computers and Electronics in Agriculture*, **69**: 33-–39. doi: 10.1016/j.compag.2009.06.011.

Baranyai, L. (2020) Laser induced diffuse reflectance imaging – Monte Carlo simulation of backscattering measured on the surface. *MethodsX*, **7**: 100958. doi: 10.1016/j.mex.2020.100958.

See Also

`MCBS` for construction of object with initial input parameters. `Setup` for initial computation of specular reflection and transport albedo. `Randomize` for adjustment of random trajectory vectors. `print` show adjusted parameters and total backscattered reflection. `Chart` for plot of calculated photon flux profile. `Export` for export of photon flux with corresponding radii.

Examples

```
## Apple simulation data according to Qin and Lu (2006).
## DOI: 10.13031/2013.20862
```

```
# create object
cfgMedia <- c(0.63,30,0,1.4)
# the 100 low number of photons is only for demonstration
cfgSimulation <- c(100,0.05,1e-9,3,0.1)
apple <- MCBS(cfgMedia,cfgSimulation)

# run simulation with default incident angle
apple <- Simulation(apple)
print(apple)

# run simulation with 10 deg incident angle
apple <- Simulation(apple,10)
Chart(apple)
```

Index