MOSE

Monte Carlo Optical Simulation Environment

Software for Photon Tracing in Biological Tissues

Ge Wang Ph.D[1]

Jie Tian Ph.D[2]

[1]CT/Micro-CT Laboratory, University of IOWA, IOWA City, U.S.A.

[2]Medial Image Processing Group, Institute of Automation, CAS, Beijing, P.P.C.

Technical Report

Release 2.0

Supported by NIH/NIBB 1R21EB001685-01

Technical Report of MOSE (V2)

MOSE is a PC-based 2D/3D simulator to predict bioluminescent signals detected by the CCD camera. Given the bioluminescent sources inside biological tissues and optical parameters of simulated biological environment, MOSE traces the propagation process of each photon (or photon packet) using Monte Carlo (MC) method. Compared with the first version of MOSE, the key improvement of the second version lies in that the input is not the simulated phantom data of animals but the real raw data scanned by computed tomography (CT). After preprocess operation of the input (e.g. image segmentation, 3D image reconstruction and surface rendering), the real biological environment can be built from the raw data of animals tissues, which is constituted by a series of triangle meshes. Except for monotonic bioluminescent sources, multi-spectral bioluminescent sources are also used to generate photons/photon packets with several kinds of energy in the second version. After the propagation during the simulated biological environment, the residual energy of each photon packet transmitting out of the whole biological tissues is recorded by the MOSE. When all the photons finish their transport, the absorption attribution schematics of the biological tissues are also easily obtained. As a whole, the simulation of photons' propagation in the second version is similar as that in the first version, including photon generation and photon transport in the simulated biological environment. However, there are completely different algorithms to realize the simulation due to the various inputs of these two versions. In the technical report of MOSE (V2), these algorithms will be introduced in detail.

1. Preprocess of Input Data

Besides the geometrical input data composed of several building blocks (e.g. cylinder, ellipsoid in 3D interface, or ellipse in 2D interface), real raw data from a series of CT slices are used as the input of the MOSE (V2). This is the critical difference between MOSE (V1) and MOSE (V2), which bring entirely various algorithms of realization in these two versions. After input, the main task of MOSE is to build the simulated biological environment, in which photon packages perform their transports. The prior knowledge of the raw data includes the image width/height of each slice, the whole numbers of all the slices, the voxel width/height, the inter-slice distance, the gray levels and the optical properties of all the biological tissues. The whole preprocess of raw data can be divided into two main steps: surface rendering and mesh simplification of each biological tissue.

Surface rendering

According to the gray levels of each tissue, it is easier to efficiently segment the target tissue from all the slices by threshold segmentation method. Then, a series of triangle meshes compose of the target tissue will be obtained by marching cubes (MC) algorithms, which is known as surface rendering. With segmentation and surface rendering algorithms, the surface of all the tissues whose grey level is between the threshold range (i.e. $TG_{\min}$ and $TG_{\max}$) can be reconstructed. Figure 1 (a) shows 2D slice of mouse thorax raw data as the input of MOSE; Figure 1 (b) and (c) respectively shows the re-sliced results including the sagittal section and the coronal section of the 3D mouse thorax. Figure 2 shows the bone surface (a), heart surface (b), and the combined surfaces of several tissues (c) obtained by surface rendering algorithms. After each 3D biological tissue is expressed by triangle meshes, the whole virtual biological environment (shown as Figure 2 (c)) is constituted by combining all the triangle meshes. These algorithms (i.e. segmentation, re-slice, surface rendering) are from the MITK (Medical Imaging ToolKit) developed by MPIG, Institute of Automation, Chinese Academy of Sciences (http://www.mitk.net).

Mesh Simplification

In general, there are more than 10^4~10^6 triangle meshes to describe a tissue of the mouse thorax after surface rendering. As we all know, the more triangle meshes are used, the more time is needed for each photon packet's propagation. Therefore, the triangle meshes should be simplified to satisfy the real-time running speed of MOSE, which is significantly important in building the virtual biological environment for photon packets' migration in real-time. In MOSE (V2), a fast mesh simplification algorithm combining half-edge data structure and modified Quadric Error Metrics (QEM) is used to simplify the original triangle meshes obtained by MC algorithm.

Data Structure. To improve the running speed of the MOSE, the mesh data structure must be convenient for searching the target mesh according to certain requirements. When half-edge mesh data structure is used, the adjacency queries between components of the mesh (e.g. vertices, faces and edges) can be quickly achieved and thus the run time is reduced remarkably. Experiments show that the initialization speed is 2 times as fast as original QEM, and the simplification speed is 2-4 times as fast as original QEM. A mesh model described by half-edge structure is shown as Figure 3.

Structures of half-edge, vertex and face in each triangle mesh is are defined as follows.

```
Struct HalfEdge
{
        Vertex * vert;          // the start vertex of the half-edge
        HalfEdge * pair;        // the adjoining half-edge of this one
        HalfEdge * next;        // the next half-edge in the same face
        Face * face;            // the face involving the half-edge
};
Struct Vertex
{
        float vcoord[3];        // the coordinates of the vertex
        float ncoord[3];        // the normal of the vertex
        HalfEdge *he;           // an arbitrary half-edge emitting from the vertex
};
Struct Face
```

```
{
    HalfEdge *he;        // an arbitrary half-edge involved in the face
};
```

Distance Function. Distance function is so important that it directly decides the collapsing sequences of all half-edges during simplification and remarkably influences the quality of the simplified results. Considering the characteristics of medical models, we integrated the information of normal vectors to the procedure of defining the distance function by adding the distance from the vertex to the tangential plane. The distance function is used to present the cost of collapsing the half-edge $\overrightarrow{P_1P_2}$:

$$DF^2(p) = \sum_i w_i D_i^2(p) + w_{P_1} D_{TP_1}^2(p) + w_{P_2} D_{TP_2}^2(p) \tag{1}$$

where $D_i(p)$ denotes the Euclidean distance between any point p and the ith mesh adjacent to half-edge $\overrightarrow{P_1P_2}$; $D_{TP_i}(p)$ denotes the Euclidean distance between the point p and the tangent planes of vertex P_i ; $w_i = S_i / \sum_i S_i$ represents the weight of the ith mesh whose area is S_i ; weight w_{P_i} (i=1,2) is unusually chosen according to the number of the triangle meshes involving the vertex P_i .

Quadric Error Metrics. In MOSE (V2), the distance function (1) based on Euclidean distance is used as the cost function of Quadric Error Metrics. Any 2D plane M can be described as $n^T u + d = 0$, where $n = [a\ b\ c]^T$ is the unit normal of the plane, $u = [x\ y\ z]^T$ presents any point inside the plane, and d is a constant. Then, the distance $D(v)$ between any point $v = [x\ y\ z]^T$ in the 3D space and plane M satisfies

$$D_M^2(v) = (n^T v + d)^2 = v^T \left(nn^T\right) v + 2(dn)^T v + d^2 . \tag{2}$$

Equation (2) can be described as

$$D_M^2(v) = v^T Q_M v = Q_M(v) , \tag{3}$$

where 4×4 matrix $Q_M = \begin{bmatrix} A & b \\ b^T & c \end{bmatrix}$ is the error matrix with respect to plane M , $A = nn^T$ is a 3×3 matrix, $b = dn$ is 1D vector, $c = d^2$ is a constant, and $v = [x\ y\ z\ 1]^T$. Considering all the triangle meshes involving vertexes P_1 and P_2 and the total distance function (1), the corresponding cost function is

$$E_Q(v) = \sum_i w_i Q_i(v) + w_{P_1} Q_{TP_1}(v) + w_{P_2} Q_{TP_2}(v) = Q(v) . \tag{4}$$

Then, the optimal point p' is calculated to minimize the cost function (4), which is the new point after collapsing the half-edge $\overrightarrow{P_1P_2}$. If matrix A is a nonsingular one, the coordinates v' of new vertex p' is usually obtained by

$$v_{new} = -A^{-1}b \tag{5}$$

satisfying a group of partial differential equation $\partial Q/\partial x = \partial Q/\partial y = \partial Q/\partial z = 0$. The cost of collapsing $\overrightarrow{P_1P_2}$ is calculated by

$$Q(v) = -b^T A^{-1} b + c \tag{6}$$

When matrix A is a singular one, there is no optimal solution to equation (4). Therefore, the coordinates v' of new vertex p' in the half-edge $\overrightarrow{P_1P_2}$ minimizing $E_Q(v')$ is chosen as the solution. If $(v_1 - v_2)^T A(v_1 - v_2) \neq 0$ and $0 \leq a \leq 1$, the optimal vertex coordinates v' is calculated by the following equations

$$\begin{aligned} v' &= a(v_1 - v_2) + v_2 \\ a &= -\frac{(v_1 - v_2)^T A v_2 + v_2^T A(v_1 - v_2) + 2b(v_1 - v_2)}{2(v_1 - v_2)^T A(v_1 - v_2)} . \end{aligned} \tag{7}$$

If the optimal vertex can't be searched, the new vertex is chosen from P_1 and P_2 which minimizes $E_Q(v')$.

When obtaining the new vertex after merging the two vertexes of each half-edge and the cost of collapsing each half-edge, we can store all the half-edges into a heap according to their costs. When collapsing that edge, the minimum cost half-edge on the heap is chosen. This method promises the minimum cost of mesh simplification and accelerates the running speed of mesh simplification.

Error Analysis. After mesh simplification, the vertexes, half-edges and faces are largely reduced. Though the running speed of MOSE is improved, the error brought by mesh simplification must be analyzed to promise the precision of the whole program. Here, we use geometry similarity metrics (GSM) to discuss the influence of mesh simplification.

The first GSM is defined as

$$E_{\max}(M_1, M_2) = \max\Big((\max_{v \in M_1} d_v(M_2), \max_{v \in M_2} d_v(M_1) \Big) \tag{7}$$

where $d_v(M) = \min_{w \in M} \| v - w \|$ denotes the distance between the vertex v and a tissue model M composed by a series of triangle meshes, $\| \bullet \|$ means the Euclidean distance of two vectors. This GSM is used to obtain the maximum error between the original tissue model M_1 and the simplified tissue model M_2. If the metrics satisfies $E_{\max}(M_1, M_2) \le \varepsilon$, all the new vertexes in M_2 generated by mesh simplification are located in the ε scope of original vertexes in the original tissue model M_1.

The second GSM is defined as

$$E_{\max}(M_1, M_2) = \frac{1}{2}\Big(\frac{1}{V_{M_1}} \sum_{v \in M_1} d_v(M_2) + \frac{1}{V_{M_2}} \sum_{v \in M_2} d_v(M_1)\Big) \tag{7}$$

where V_M is the number of the vertexes in tissue model M. This GSM presents the average error between the original tissue model M_1 and the simplified tissue model M_2.

When the original tissue model M_1 with triangle meshes is a convex, the third GSM is defined according to the volume change between M_1 and the simplified tissue model M_2. Take all the coordinates of the vertexes and the center R of the bounding box as the prior knowledge. It is easier to calculate the volume of each tissue model M by the sum of all tetrahedrons whose vertexes are the three vertexes of one triangle mesh and the center R.

InorOut Function

The InorOut function is frequently used when running simulation. When a photon is generated, we call this function to compute which tissue it lies in. And this work must be repeated many times through the whole process of a photon's propagation, because after each step of photon's movement, the information about its position must be updated. In general, a photon moves about 300 steps before it is terminated, and 10^6 photons are traced when running simulation. Therefore, the correctness and efficiency of InorOut function will significantly influence the performance of MOSE. Different strategies are applied to different situation to achieve faster speed with high accuracy.

InorOut Strategy For Basic Shapes. In earlier version of MOSE, the shape of light source and biologic tissues are represented by a combination of the basic geometric forms, such as the circle, ellipse, rectangle in two-dimensions, and the sphere, ellipsoid, cylinder in three-dimensions. Consequently, we can easily test whether a photon is inside an organ through comparing its position with the organ's boundary which can be given by mathematical expressions. This strategy is the most efficient one within all we introduce.

InorOut Strategy For 2D Polygons. In MOSE V2, polygons formed by lines and arcs are taken as the basic elements for representation. Though much more intricate shapes can be described, the InorOut strategy for basic shapes no longer works. Therefore, a new strategy with higher adaptability is introduced. This strategy is based on the *Jordan Curve Theorem*. Essentially, it says that a point is inside a polygon if, for any ray from this point, there is an odd number of crossings of the ray with the polygon's edges. We call this algorithm the crossings test. Optimizations can be done as follows:

1. Consider the test point to be at the origin and shoot a ray from it along the +X axis.
2. Check the edges against this point, if the Y components of a polygon edge differ in sign, then the edge can cross the test ray.
3. In this case, if both X components are positive, the edge and ray must intersect and a crossing is recorded.
4. Else, if the X signs differ, then the X intersection of the edge and the ray is computed and if positive a crossing is recorded.

Figure 4 shows the crossings test algorithm.

When the test ray intersects one or more vertices of the polygon, the crossing point at the vertex will be counted twice, and the test will turn out to be failure. Different situations of vertices intersection are shown in Figure 5. This problem can be resolved by considering that whenever the ray would intersect a vertex, the vertex is always classified as being infinitesimally above the ray. In this way, no vertices are intersected and the running speed is speedier.

InorOut Strategy For 3D Polyhedrons. Real raw data from a series of CT slices are used as the input in MOSE V2, and after surface rendering and simplification process, triangle meshes are generated to formed polyhedrons which represent the tissues. Here, we extend the InorOut strategy from two-dimension to three-dimension based on the same theory.

1. Run a semi-infinite ray up (e.g., along the +Z axis) from the point **P**.
2. If **P** lies below the plane of **F**, project both **P** and **F** onto the plane of **XOY**
3. Do the 2D point-containment test.

If the ray intersects an edge/a vertex, we perturb **P** slightly and apply InorOut strategy again. Then if the meshes that share the same edge/vertex form a smooth surface (shown as Figure 5(a)), **P** must cross only one of them, on the other hand, if them form a protruding shape (shown as Figure 5(b)), **P** must get even crossings.

2. Photon Generation

There are two main types of bioluminescent sources (BLS) expressed by a series of triangle meshes in MOSE (V2). One is numbers of regular sources (RS) located in certain 3D geometric areas (e.g. sphere, cylinder), where these sources conform to given probability distribution functions (e.g. uniform, normal). The other is numbers of artificial sources (AS) which are in 3D irregular areas generated by interactively modifying RS area according to users' purpose. Position-sample and angle-sample are the two requisite steps, respectively deciding the incident position and the propagation direction of photon packets. Given the distribution functions of BLS, the Monte Carlo method is used to randomly sample the incident position, the incident direction of photon packets, and incident energy of photon packets. In the following sections, it is assume that these BLS are uniformly distributed in certain 3D areas for brevity.

Position-sample

With the triangle meshes describing the whole surface of the BLS area, it is easier to obtain the bounding box of this area (i.e. $x_{\max}, x_{\min}, y_{\max}, y_{\min}, z_{\max}, z_{\min}$). According to the uniform distribution, the direct sampling of photon position used by the Monte Carlo method is

$$\begin{cases} x = x_{\min} + \xi_1 (x_{\max} - x_{\min}) \\ y = y_{\min} + \xi_2 (y_{\max} - y_{\min}) \\ z = z_{\min} + \xi_3 (z_{\max} - z_{\min}) \\ InorOut(x, y, z) = 1 \end{cases},$$

where ξ_i is pseudo-random number which is uniformly distributed over the interval $(0,1)$, and function $InorOut(x, y, z) = \begin{cases} 0 & in \\ 1 & out \\ 2 & on\ the\ boundary \end{cases}$ is used to judge whether point (x, y, z) is inside the BLS area bounded by triangle meshes.

Direction-sample (angle-sample)

The azimuthal angle φ and deflection angle θ are the two critical parameters of the dynamic spherical coordinate system, describing the propagation direction of any photon packet. In MOSE (V2), azimuthal angle φ is uniformly distributed between 0 and 2π, so the direction-sample is given by

$$\varphi = 2\pi\xi_\varphi ,$$

where ξ_φ is pseudo-random number which is uniformly distributed over the interval zero to one.

The cosine of deflection angle θ is uniformly distributed between 0 and 1, so the sample of $\cos\theta$ is calculated by

$$\cos\theta = 2\xi_\theta - 1 ,$$

where ξ_θ is pseudo-random number which is uniformly distributed over the interval zero to one.

The direction of travel with three directional cosines is specified by taking the cosine of the angle that the photon's direction makes with each axis. The direction (μ_x, μ_y, μ_z) is computed by

$$\begin{cases} \mu_x = \sin\theta\sin\varphi \\ \mu_y = \sin\theta\cos\varphi \\ \mu_z = \cos\theta \end{cases} .$$

Energy-sample

Because the tissue optical properties are wavelength/energy dependent, we will split the spectrum (500–760 nm) of multi-spectral BLS into N (e.g. N=32) consecutive segments during the Monte Carlo simulations. Photon packets generated from the same segment is assumed to have the same wavelength/energy. As all the energy levels are uniformly distributed, there are two ways to complete the simulations of multi-spectral BLS. One is that photon packets of each energy level are respectively simulated by MOSE; the other is that energy sample is used to obtain the energy level of each photon packet. After propagations of all photon packets terminate, the transmission and absorption values are recorded according to various energy levels. According to the uniform distribution and MC method, the energy level can be sampled by $E = Int(32\xi_E)$, where ξ_E is pseudo-random number which is uniformly distributed over the interval zero to one, and *Int(x)* returns the largest integer that is less than or equal to *x*.

3. Photon propagation in biological tissues

3.1 Photon Move To Next Interaction Site

An efficient method chooses a variable step size for each photon step. The step size of photon packet is calculated based on the randomly sampling as follows:

$$\Delta s = \frac{-\ln\xi}{\mu_a + \mu_s},$$

where μ_a, μ_s are absorption coefficient and scattering coefficient of biological tissues respectively.

In MOSE (V2), the spatial position and propagation direction are critical variables of each photon packet, which record the trace of photon migration. It is convenient to describe the photon's spatial position with the Cartesian coordinates (x, y, z) and the directional cosines of travel (μ_x, μ_y, μ_z) specified by the cosine of the angle between transport direction of photons and the three coordinate axes. For a photon located at (x, y, z), the relationship between traveling distance Δs in the direction (μ_x, μ_y, μ_z) and the new interaction site (x', y', z') is given by

$$\begin{cases} x' = x + \mu_x \Delta s \\ y' = y + \mu_y \Delta s \\ z' = z + \mu_z \Delta s \end{cases}.$$

3.2 Photon Absorption

The technique of implicit capture assigns a weight, equal to unity, to each photon as it enters tissue. After each propagation step, the photon packet is split into two parts: a fraction is absorbed and the rest is scattered. The fraction of photon packet absorbed is

$$a = \Delta w = \frac{\mu_a}{\mu_a + \mu_s} = 1 - \frac{\mu_s}{\mu_a + \mu_s} = 1 - \alpha$$

where α is the single particle albedo. One part of the photon weight Δw is stored into absorption matrix $A\_xyz(i_x, i_y, i_z)$ in MOSE at the local grid element. Accordingly, the new photon weight w' is given by $w' = \alpha w$, which is the fraction of the photon packet scattered.

3.3 Photon Termination

There are two ways for photon packets stopping propagation. On the one hand, photon packets are totally absorbed by tissues; on the other hand, photon packets reach CCD camera and are absorbed by detectors. When photons terminate, the flag named “dead” is set to 1. Otherwise, it is 0, which means photon is alive now. A technique named Russian roulette is used to terminate a photon packet once its weight drops below a specified threshold (e.g. 0.001). Russian roulette gives the packet one chance in m (e.g. m=10) of surviving with a weight of *mw*. If the packet does not survive the Russian roulette, the photon weight is reduced to 0 and the photon is terminated. Russian roulette is expressed by formula as follows:

$$w = \begin{cases} mw & if\ \xi \le 1/m \\ 0 & others \end{cases},$$

where ξ is the uniformly init pseudo random number.

3.4 Photon Scattering

In MOSE (V2), a normalized phase function describes the probability density function for the azimuthal and longitudinal angles of a photon packet when it is scattered. Azimuthal angle φ is uniformly distributed between 0 and 2π, so the direction-sample is given by $\varphi = 2\pi\xi_\varphi$. The probability distribution for the cosine of deflection angle $\cos\theta$ is described by Henyey-Greenstein function

$$p(\cos\theta) = \frac{1-g^2}{2(1+g^2-2g\cos\theta)^{3/2}},$$

so the direct sample of $\cos\theta$ is calculated:

$$\cos\theta = \begin{cases} \frac{1}{2g}(1+g^2-(\frac{1-g^2}{1-g+2g\xi_\theta})^2) & if\ g \neq 0 \\ 2\xi_\theta - 1 & others \end{cases},$$

where g is the anisotropy factor of the medium. ξ_φ and ξ_θ are pseudo-random numbers uniformly distributed over the interval zero to one.

In 3D MOSE (V2), with the moving spherical coordinate system and the transform between coordinates system, the new propagation direction of photon packet (μ'_x, μ'_y, μ'_z) is computed by

$$1. |\mu_z| < 0.99999$$

$$\begin{cases} \mu'_x = \sin\theta(\mu_x\mu_z\cos\varphi - \mu_y\sin\varphi)/\sqrt{1-\mu_z^2} + \mu_x\cos\theta \\ \mu'_y = \sin\theta(\mu_y\mu_z\cos\varphi + \mu_x\sin\varphi)/\sqrt{1-\mu_z^2} + \mu_y\cos\theta \\ \mu'_z = -\sin\theta\cos\varphi\sqrt{1-\mu_z^2} + \mu_z\cos\theta \end{cases}$$

$$2. |\mu_z| \geq 0.99999$$

$$\begin{cases} \mu'_x = \sin\theta\cos\varphi \\ \mu'_y = \sin\theta\sin\varphi \\ \mu'_z = SIGN(\mu_z)\cos\theta \end{cases}$$

where $SIGN(x) = \begin{cases} 1 & if\ x \geq 0 \\ -1 & others \end{cases}$.

3.5 Photon Hit The Boundary Of Tissues

During the transportation of a photon packet, it may hit the boundary of the current tissue, where the boundary may either be an interface between the tissue and the ambient medium or an interface between the current tissue and another adjoining tissue. The photon packet can be either internally reflected back into the current tissue by the boundary or transmit across the boundary into the adjoining tissue. Then, the photon propagation will continue. When the photon packet escapes all biological tissues to the ambient medium, it will be observed as internal reflectance or transmittance depending on the incidence angle onto the boundary. In the following section, we compute critical angle α_c, the cosine of incident angle α_i, cosine of transmission angle α_t, direction of photon packets before and after hitting the boundary.

Incident Angle

After preprocess of the input data, each biological tissue can be described by a series of triangle meshes. Three vertices' coordinates (i.e. $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, and $P_3(x_3, y_3, z_3)$) and the normal N of each triangle mesh are stored into a matrix, which can efficiently accelerate the running speed of MOSE. If the photon packet whose propagation direction is expressed as I hits the boundary triangle mesh at point P, the cosine of photon's incident angle at point P is given by $\cos\alpha_i = \left|\frac{I \bullet N}{|I| \times |N|}\right|$.

Transmission Angle

Snell's law indicates the relationship between the angle of incident α_i, the angle of transmission α_t, the refractive indices of the media that the photon packet is incident from, n_i, and transmitted to, n_t:

$$n_i \sin\alpha_i = n_t \sin\alpha_t .$$

The cosine of transmission angle is calculated by:

$$\cos\alpha_t = \begin{cases} \cos\alpha_i & if\ n_i = n_t \\ (1 - \sin^2\alpha_t)^{1/2} & others \end{cases}$$

Critical Angle

Before the photon packet is decided whether it is internally reflected or transmitted, several parameters must be calculated such as critical angle α_c and $R(\alpha_i)$.

Critical angle (possible only when $n_i > n_t$) is calculated by formulas as follows:

$$\alpha_c = \begin{cases} \sin^{-1}(n_i / n_t) & if\ n_i > n_t \\ 0 & others \end{cases},$$

where n_i, n_t are refractive indices of media which photon is incident from and transmitted to respectively. If α_i is larger than the critical angle, $R(\alpha_i) = 1$. Otherwise, $R(\alpha_i)$ is calculated by Fresnel's formulas:

$$R(\alpha_i) = \begin{cases} \frac{1}{2}\left[\frac{\sin^2(\alpha_i - \alpha_t)}{\sin^2(\alpha_i + \alpha_t)} + \frac{\tan^2(\alpha_i - \alpha_t)}{\tan^2(\alpha_i + \alpha_t)}\right], & \text{if } \alpha_i \le \alpha_c; \\ 1, & \text{others.} \end{cases},$$

where α_i, α_t are incident angle and transmission angle at the boundary respectively. If $\xi \le R(\alpha_i)$, the photon packet is internally reflected. If $\xi > R(\alpha_i)$, the photon packet transmits.

Internally reflected

Take example for internal reflection at boundaries of tissues in 3D MOSE. If the photon packet is internally reflected, the photon packet stays at the boundary and its directional cosine (μ_x, μ_y, μ_z) must be changed. The geometry of photon internal reflection is shown as Figure 4. (x, y, z) are the coordinates of photon hitting the boundary. Vectors $I(\mu_x, \mu_y, \mu_z)$ and $R(\mu'_x, \mu'_y, \mu'_z)$ describe incident direction cosines and internally reflected direction cosines respectively. Vector $N(\mu_{nx}, \mu_{ny}, \mu_{nz})$ expresses the direction of outside normal at point $P(x, y, z)$. Through the vectors schematic, we can get the equation as follows:

$$R = I - 2\frac{I \bullet N}{|I||N|}\frac{N}{|N|}.$$

Therefore, the direction cosines of internally reflected photons (μ'_x, μ'_y, μ'_z) are given by

$$\begin{cases} \mu'_x = \mu_x - k\mu_{nx} \\ \mu'_y = \mu_y - k\mu_{ny} \\ \mu'_z = \mu_z - k\mu_{nz} \end{cases},$$

where $k = \frac{2I \bullet N}{|N|^2}$.

Transmitted

Take example for transmittance at boundaries of tissues in 3D MOSE. If the photon packet transmits from the tissues to outside medium, the directional cosines of photon packet are changed from vector $I(\mu_x, \mu_y, \mu_z)$ to vector $T(\mu_x^T, \mu_y^T, \mu_z^T)$. α_t is the transmittance angle of photon packet. Other parameters are the same as what the internally reflected defines. The geometry of photon transmission is shown as Figure 4. Through the vectors schematic, we can get the equation as follows

$$T = \sin\alpha_t \cdot \frac{J}{|J|} + SIGN(I \bullet N) \cdot \cos\alpha_t \cdot \frac{N}{|N|}, \qquad (4\text{-}24)$$

where

$$J = I - \frac{I \bullet N}{|N|} \cdot \frac{N}{|N|}, \qquad (4\text{-}25)$$

$$SIGN(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \qquad (4\text{-}26)$$

$(\mu_x^T, \mu_y^T, \mu_z^T)$ may also be easily calculated by those above formulae according to the attributions of vector computations. When the photon packet reaches a triangle mesh whose index is *i*, the residual photon weight of it will be stored into the transmittance matrix $T_{mesh}(i)$ by the equation $T_{mesh}(i) = T_{mesh}(i) + w$. After all photon packets terminate their propagations, the matrix T_{mesh} will present the transmittance distribution on the boundary of the whole biological tissue surface.

3.6 Scored Physical Quantities

Internal photon distribution

During the simulation, absorbed photon weights are scored into the absorption array $A_{xyz}(i_x, i_y, i_z)$, where i_x, i_y, i_z are the indices for grid elements of x, y, z axes. The raw data $A_{xyz}(i_x, i_y, i_z)$ provides total weight in each grid element in 3D grid system. The total weight absorbed in the tissues is computed as follows:

$$A = \sum_{i_x=0}^{N_x-1} \sum_{i_y=0}^{N_y-1} \sum_{i_z=0}^{N_z-1} A_{xyz}(i_x, i_y, i_z).$$

Then, all these quantities have been scaled appropriately to get the densities:

$$\begin{aligned} &A_{xyz}(i_x, i_y, i_z) = A_{xyz}(i_x, i_y, i_z)/(dxdydzN) \\ &A = A/N \end{aligned},$$

where dx, dy, dz are grid separations of x, y, z axes respectively, N is the total number of photon packets. Given the total energy of all incident photon packets $E_{incident}$, the total energy absorbed by biological tissues is calculated:

$$E_A = A \times E_{incident}$$

Transmittance

After tracing total number of photon packets (N), the raw $T_{mesh}(i)$ offers the total photon weight in each grid element in 3D grid system. The total transmittance is computed by

$$T = \sum_{i_x=0}^{N_x-1} \sum_{i_y=0}^{N_y-1} \sum_{i_z=0}^{N_z-1} T_{mesh}(i)$$

Then, all these quantities have been scaled appropriately to get the densities:

$$\begin{aligned} &T_{mesh}(i) = T_{mesh}(i)/(dxdydzN) \\ &T = T/N \end{aligned},$$

where dx, dy, dz are grid separations of x, y, Z axes respectively. Given the total energy of all incident photon packets $E_{incident}$, the total energy emitting out of the biological tissues is calculated:

$$E_T = T \times E_{incident}.$$

With the structure and geometric parameters of CCD camera, the energy absorbed by detectors can be easily computed.

3.7 Estimation of MOSE Precision

The results of MOSE represent an average of the contribution from many histories sampled during the course of all the photons propagation. An important quantity equal in stature to the results of MOSE is the statistical error. From it, we not only can gain insight into the quality of the result, but also can determine if a tally is well behaved. This section tells us how to calculate the estimated means and relative errors of the results of MOSE in order to ensure a well-behaved and meaningful tally.

Suppose $p(x)$ is the probability density function of the random variable x being estimated. The true answer (or mean) $E(x)$ is the expected value of x, whose function is $E(x) = \bar{x} = \int xp(x)dx$. The quantity $E(x)$ is seldom known because $p(x)$ is not known directly but can be estimated by Monte Carlo through the random walk process as $\bar{x}$, which is given by

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

where x_i is the value of x sampled from $p(x)$ for the i^th history and N is the number of histories calculated. The Strong Law of Large Numbers states that if $E(x)$ is finite, $\bar{x}$ tends to the limit $E(x)$ as N approaches infinity.

The variance of the population of x is given by

$$o^2 = \int (x - E(x))^2 p(x)dx = E(x^2) - (E(x))^2 \ .$$

The square root of the variance is o, which is the standard deviation of the population of scores. Though o is seldom known, it can be estimated as S, given by (for large N)

$$S^2 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1} \sim \overline{x^2} - \bar{x}^2 \quad \text{where}$$

$$\overline{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^{\ 2} \ .$$

The quantity S is the estimated standard deviation of x based on the values of x_i that were actually sampled. The estimated variance and the estimated standard deviation of $\bar{\text{x}}$ are respectively given by the following formulas:

$$S_{\bar{x}}^2 = \frac{S^2}{N} \qquad\qquad S_{\bar{x}} = \sqrt{\frac{\overline{x^2} - \bar{x}^2}{N}} \ .$$

Then the relative error in MOSE can be calculated:

$$R = \frac{S_{\bar{\text{x}}}}{\bar{\text{x}}} \ .$$

It is important to note that $S_{\bar{x}}$ is proportional to $1/\sqrt{N}$, which is the inherent drawback to the Monte Carlo method. For example, to halve $S_{\bar{x}}$, four times the original number of histories must be calculated. Hence, a large number of histories and a large amount of computation time must be needed to get a more accurate result from Monte Carlo.

4. Graphic Editing Tools

Superquadrics

The superquadric models were introduced into the computer graphics field in 1981. There are four types of superquadric models: superellipsoid, supertoroid, and superhyperboloid with one or two sheets. Among these four types, only the superellipsoid defines a closed surface without holes, which is always consistent with the condition of real environment. Therefore, the superellipsoid is commonly referred to as the superquadric. In our simulation platform, the superellipsoid are used as a particular building block.

A superellipsoid surface is defined by an implicit equation:

$$(|x/r_x|^{2/\varepsilon_2} + |y/r_y|^{2/\varepsilon_2})^{\varepsilon_2/\varepsilon_1} + |z/r_z|^{2/\varepsilon_1} = 1,$$

where radius parameters r_x, r_y, r_z denote the scaling factors on x, y and z axes, squareness parameters $\varepsilon_1, \varepsilon_2$ are the shape parameters related to the squareness/roundness/pinchedness in the longitudinal and horizontal directions respectively. Another concise definition of the superellipsoid in the sphere coordinate system is described as

$$r(\eta, \omega) = \begin{bmatrix} r_x \operatorname{sgn}(\cos\eta\cos\omega)|\cos\eta|^{\varepsilon_1}|\cos\omega|^{\varepsilon_2} \\ r_y \operatorname{sgn}(\cos\eta\sin\omega)|\cos\eta|^{\varepsilon_1}|\cos\omega|^{\varepsilon_2} \\ r_z \operatorname{sgn}(\sin\eta)|\sin\eta|^{\varepsilon_1} \end{bmatrix},$$

where η, ω are latitude and longitude angles with $-\pi/2 \le \eta \le \pi/2$, $-\pi \le \omega \le \pi$, and $\operatorname{sgn}(x)$ is the sign function. By varying the value squareness parameters $\varepsilon_1, \varepsilon_2$, a wide range of shapes can be conveniently generated. Roughly speaking, if a squareness parameter is significantly less than 1, the geometry is somewhat square; if it is close to 1, the object is quite round; if it is close to 2, the shape has a flat bevel; if it is greater than 2, the structure is pinched. Figure 5 presents a series of the shapes that can be obtained from various combinations of squareness parameters. With certain shape parameters $\varepsilon_1, \varepsilon_2$, the superellipsoid can be transferred into regular geometric graphics. For instance, if $\varepsilon_1 = 0, \varepsilon_2 = 1$, the superellipsoid is a cylinder; if $\varepsilon_1 = 1, \varepsilon_2 = 1$, the superellipsoid is a ellipsoid.

When the building blocks are the superellipsoid, it is easy to judge whether a point $p(x, y, z)$ is inside of the shape according to the inside-outside function:

$$F(x, y, z) = (|x/r_x|^{2/\varepsilon_2} + |y/r_y|^{2/\varepsilon_2})^{\varepsilon_2/\varepsilon_1} + |z/r_z|^{2/\varepsilon_1}.$$

If function $F(x, y, z) < 1$, point p is inside the superellipsoid; if $F(x, y, z) = 1$, p is on the boundary of the superellipsoid; if $F(x, y, z) > 1$, p is inside the superellipsoid.

2D interactive graphic editing tool

In 2D simulation environment, any 2D irregular bioluminescent source can be described by 2D interactive graphic editing tools and can be conveniently modified by interactive operations. Shown as Figure 6, any 2D shape is determined by the position relationship of all vertexes (i.e. small solid circles on the contour). After several original vertexes are given according to the requirements or the prior knowledge of operators, the original contour (Figure 6 (a)) of the bioluminescent source is formed by smoothly connecting the adjoining vertexes with polynomial curves or arcs. Then, the original contour can be conveniently modified by adding, deleting, and moving vertexes (Figure 6 (b) and (c)). For example, if one part of the local contour is smooth, several vertexes are enough to depict its fine features; if it is rough (e.g. sharp protuberances and hollows), more vertexes are needed to describe the details of the local contour, shown as Figure 6 (d). Given a point P, the line $\overrightarrow{L_P}$ with certain direction can be easily obtained. The number N of intersections between $\overrightarrow{L_P}$ and the contour described by 2D interactive GET is used to determine whether the point P is inside of the contour. If N is odd, the point P is inside the shape; if N is even or zero, the point P is outside the shape; if the point P satisfies the function of contour formed by polynomial curves or arcs, it is on the boundary of the shape.

3D interactive graphic editing tool

The original shape of any 3D irregular object is usually chosen as a sphere or a cylinder whose parameters are determined according to the prior knowledge of operators. In our simulation platform, the default original shape of the 3D bioluminescent source is a sphere, and the Bezier cubic spline mode is applied as the interactive editing tool. First, the volume of interest (VOI) of the original object is selected, which defines the local surface to be modified. Then, the selected local surface of the original 3D object can be dragged via a so-called control point p_c along any direction, which can be repositioned interactively. The position of the original control point p_c is determined by the shape of the local surface S_L which needs to be modified. When the position of control point is changed, the whole local surface is modified accordingly by a Bezier cubic spline mode.

In 3D simulation environment, multiple Bezier cubic splines are used to describe the modified local surface inside the VOI of the 3D object. To make this 3D interactive mode easier to be understood, we begin with the 2D Bezier cubic spline curve. The 2D Bezier cubic spline curve can be described algebraically by a Bernstein polynomial of degree 3[10]:

$$p(t) = (1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3 ,$$

where t varies between *0* and *1*. Figure 7 shows the 2D typical Bezier cubic splines. From the equation (4), it is evident that four points p_0, p_1, p_2, and p_3 are needed to determine the shape of the spline curve. The two end points p_0 and p_3 (shown as Figure 7) are fixed, because they are the boundary points of the selected local surface of the original object. However, it is still difficult to interactively operate on the other two points p_1 and p_2 simultaneously. The solution is to search one control point p_c to express two points p_1 and p_2. In our simulation platform, we chose points p_1 and p_2 are the midpoints of line segments $p_0 p_c$ and $p_3 p_c$ respectively. Then, equation (4) can be rewritten as:

$$p(t) = 1.5[(1-t)^2 t + (1-t)t^2] p_c + [(1-t)^3 + 1.5(1-t)^2 t] p_0 + [t^3 + 1.5(1-t)t^2] p_3 .$$

Because the two points p_0 and p_3 are known, the whole Bezier cubic spline can be determined by the control point p_c only. When the control point is moved, the local surface S_L can be conveniently modified accordingly (shown as Figure 7 (b)).

The mechanism of 3D surface modification is the same as that of 2D surface modification. The difference lies in that 3D local surface is made up of many Bezier cubic splines with the same control point p_c. Known the 3D original local surface to be modified and a series of initial Bezier cubic splines determined by the control point, the 3D local surface of the object is modified by a group of Bezier cubic splines in 3D simulation environment (shown as Figure 8).

In our 3D simulation platform, all the 3D shape including the biological tissues and bioluminescent sources are described by a series of triangle meshes. It is crucial to judge whether a point is inside a 3D irregular shape. Given a point P and a certain irregular contour, the line $\overrightarrow{L_P}$ with certain direction can be easily obtained. The number N of intersections between $\overrightarrow{L_P}$ and the contour described by 3D interactive graphic editing tools is used to determine whether the point P is inside of the contour. If N is odd, the point P is inside the shape; if N is even or zero, the point P is outside the shape; if the point P satisfies the function of contour composed of many triangle meshes, it is on the boundary of the 3D shape.

Results with interactive graphic editing tools

All the interactive graphic editing tools presented above have been integrated into our simulation platform, with which the irregular 2D/3D shapes can be conveniently generated and modified according to the interactive operations and prior knowledge of the operator. Take example for the modification of 3D bioluminescent source. The input of our simulation platform is a series of slices obtained from micro-CT. After segmentation, surface rendering and mesh simplification, the virtual biological environment can be built from the input data. For example, the purple graphics indicates the lib and heart of the mouse thorax with transparent effect in Figure 9. When the virtual biological environment is built, a control panel is used to give the parameters of 3D original bioluminescent source, such as, the coordinates of the center, the scaling factors on x, y and z axes, the original shape and distribution function of the object. With the movement of slides, the position and dimensions of the bioluminescent source can be easily modified. Figure 9 (a) shows the biological tissues and the original shape of 3D bioluminescent sources; Figure 9 (b) presents the movement of the source generated by the modification of the parameters; and Figure 9 (c) denotes the change of source dimensions.

The 3D local surface of the bioluminescent source can be easily modified by the Bezier cubic spline mode. In Figure 10, the red sphere is the original bioluminescent source and the yellow local surfaces are generated by Bezier cubic splines.

5. Software Design

Functions and Interface

Compared with MOSE (V1), MOSE (V2) has more built-in simulation functions and a friendlier interface handling input parameters, output files, and display. In particular, the simulation of real biological tissue environment has been realized and integrated into the same program framework as 2D/3D simulation with geometric building blocks in MOSE (V2). These three modes can be switched by pressing two buttons on the interface conveniently. MOSE (V2) implements all the functions described in the above sections via a user-friendly interface created with the Visual C++ programming language and OpenGL techniques.

After preprocess of the volume data obtained from micro-CT, each biological tissue and simulated biological environment can be clearly displayed according to operators' command. And the bioluminescent sources inside biological tissues are added in the simulated biological environment. With the control panel of the interface (shown in Figure 9) and the 3D interactive graphic editing tools (introduced in Section 4), operators can easily modify shapes of the original sources formed by 3D geometric building blocks (e.g. sphere, cylinder). Before the simulation, one can choose the display mode of the photon propagation in biological environment: tracing all the photon propagation paths; only tracing those photons that reach detectors; only tracing the photons with selected indexes. We can then observe the photon transport through the biological tissues in real-time. The transport paths of different photon packages are highlighted in different colors. While tracing photon packets, the operator may stop and restart the display of tracing by pressing a switch button at any moment. Once the simulation is finished, we can retrieve output files recording the absorption data, transmission data, running time, and so on. When a pseudo color scheme is chosen, the distribution maps of absorption and transmission can be graphically displayed as needed.

Search Strategy

In general, the simulated biological environment is quite complicated, because there are several tissues described by a series of triangle meshes. As a result, given a point inside the simulated biological environment, it is usually time-consuming to find the right tissue containing the point. Two type of table search strategy were utilized to speed up the process.

The first table search is usually performed to judge the location of any bioluminescent photon. After preprocess of the volume data obtained from micro-CT, the bounding-box of each biological tissue is calculated, which includes the minimum values and maximum values of the tissue along three axes of Cartesian coordinates. According to these building-boxes and the indexes of tissues, three tables T_1, and T_2 are generated respectively. With the pre-specified separations along the Cartesian coordinates, the whole simulated biological environment can be divided into a series of voxeles. Table T_1 is a 3D matrix with the same size of the voxel matrix, whose values are the indexes of biological tissues containing the voxel.

The second table search is used in InOrOut function. Moreover, it is also applied when photon hitting any tissue boundary during its propagation in biological environment. Because each biological tissue is made up of a series of triangle meshes, it is tremendously time-consuming to calculate whether the photon trace and each triangle mesh intersect. Table T_2 offers a fine strategy to reduce the index range to be searched when calculating the intersection on the boundary. To quick search the intersection between the photon trace and the biological tissue, all triangle meshes of each biological tissue can be divided into M regions. All indexes of triangle meshes in each region are saved into Table $T_2(i)$ $1 \le i \le M$. Then, if the photon hits the tissue boundary in Region j, only the triangle meshes whose indexes are recorded in $T_2(j)$ will be used to calculate the intersection. With the result of table search, the number of triangle meshes needed to be searched is remarkably reduced. Then, with three vertexes of each triangle mesh, it is easy to obtain the intersection on the boundary. As a result, the running speed of dealing with boundary effect is largely improved.