Lattice-Based Volumetric Global Illumination

Feng Qiu, Fang Xu, Zhe Fan, Neophytou Neophytos,
Arie Kaufman, *Fellow, IEEE*, and Klaus Mueller, *Senior Member, IEEE*

Abstract—We describe a novel volumetric global illumination framework based on the Face-Centered Cubic (FCC) lattice. An FCC lattice has important advantages over a Cartesian lattice. It has higher packing density in the frequency domain, which translates to better sampling efficiency. Furthermore, it has the maximal possible kissing number (equivalent to the number of nearest neighbors of each site), which provides optimal 3D angular discretization among all lattices. We employ a new two-pass (illumination and rendering) global illumination scheme on an FCC lattice. This scheme exploits the angular discretization to greatly simplify the computation in multiple scattering and to minimize illumination information storage. The GPU has been utilized to further accelerate the rendering stage. We demonstrate our new framework with participating media and volume rendering with multiple scattering, where both are significantly faster than traditional techniques with comparable quality.

Index Terms—Volume visualization, volume rendering, participating media, lattice, FCC lattice, sampling, multiple scattering, GPU.

1 Introduction

Direct volume rendering has been an important technique in the graphics and visualization community for many years. Usually the volume to be rendered is a function defined on 3D (or higher dimension) space and discretized into a set of cells or points. Here, a regular discretization has a single generating matrix, which is called a lattice [6]. Mathematically, a lattice in R^n is a discrete subgroup [9] of R^n that can be generated from a vector basis by a linear combination with integral coefficients. In other words, a lattice is the subgroup $a_1v_1 + a_2v_2 + \ldots + a_nv_n$, where $v_1, v_2, \ldots, v_n$ is the vector basis, v_i is the generation vector, and a_i are integers. The sites of a lattice are connected with a set of lines (or links). A site is defined as a neighbor of another site if they are connected by one lattice link. The lattice sites result from the discretization of space or space-time, and the lattice links discretize the angular space. The most frequently used lattice for volume visualization is the Cubic Cartesian (CC) lattice with generation vectors of $X = (1,0,0)$, $Y = (0,1,0)$, and $Z = (0,0,1)$. Other grids such as Body-Centered Cubic (BCC) and Face-Centered Cubic (FCC) lattices have also been investigated [22, 24, 29].

Direct volume rendering algorithms reconstruct a continuous function, which is projected to a 2D image. This procedure involves dimension reduction thus inevitably loses some information. To capture more details, many lighting and illumination methods have been developed. Local illumination models omit sophisticated effects such as multiple scattering and indirect illumination for the sake of rendering speed. However, they are the dominant light-object interaction for participating media (smoke, clouds) and many translucent materials. To cope with these effects, volumetric global illumination techniques are required in order to present important visual features [28].

Global illumination has not been widely employed in volume rendering because of the computation complexity. When a photon encounters an object, it might be reflected, refracted, and scattered many times before finally being absorbed or exiting the scene. It is closely related to the radiative transfer problem that has been studied by physicists for decades [5]. The simulation of all kinds of interaction is time-consuming, and many simplified models have been proposed in computer graphics. Max [21] has evaluated several optical models for direct volume rendering and presented an integral equation for light transport in volumes, which includes multiple scattering. Blinn [2] has analytically solved the transport equation for constant density medium with single scattering. Kajiya and von Herzen [17] have proposed tracing rays in inhomogeneous volumes. To calculate multiple scattering, spherical harmonics have been used. Radiosity uses finite element methods to solve the transport equation. Rushmeier and Torrance [27] have exploited the zonal method for isotropic scattering. Max [20] has extended the discrete ordinates method to capture anisotropic multiple scattering. Hegeman et al. [14] have proposed a two-pass approach for strong forward scattering. Harris and Lastra [13] have used a similar approach to render clouds. Kniss et al. [18] have introduced a volume lighting model for GPU-accelerated volume rendering with forward scattering using a single pass algorithm based on half angle slicing. Riley et al. [26] have extended this method to render atmospheric phenomena. Jensen [16] has presented the photon mapping method for rendering participating media. Cerezo et al. [3] have provided a nice survey on rendering participating media. Finally, Geist et al. [12] have revised the Lattice-Boltzmann method to render participating media.

In this paper, we describe a new volumetric global illumination framework, which exploits both spatial and angular discretization on lattices. In computer graphics, spatial discretization has been well-studied to simplify the calculation of light-object interaction, but angular discretization has not been fully exploited. Specifically, we adopt the FCC lattice because it has better sampling efficiency compared with the CC lattice and it provides optimal angular discretization among all lattices. We further describe a new two-pass algorithm to render participating media and volumes with multiple scattering. The idea of this algorithm is that the traced photons only move along the lattice links. We call these photons “diffuse photons”. Here, the phase function is discretized on the lattice links to simplify the diffuse photon tracing and radiance estimation. The storage of diffuse photons is minimized by storing the number of photons on lattice links. For flexibility and extensibility, we also implemented tracing photons with accurate direction, which are called “specular photons” in this paper. The O-Buffer data structure proposed by Qu et al. [25] has been exploited to reduce the storage space of specular photons. Our volumetric global illumination framework is capable of producing high quality images and is significantly faster than traditional methods. This general and flexible framework can be extended to render hybrid scenes with both volumes and surface objects.

Our paper is structured as follows. The definition and mathematical properties of FCC lattices are explained in Section 2. Section 3 analyzes the sampling scheme on FCC lattices. Section 4 and 5 present the rendering methods and implementation details, respectively. The rendering results are shown in Section 6. Finally, we conclude the paper and describe future work.

- *Feng Qiu, Fang Xu, Zhe Fan, Neophytou Neophytos, Arie Kaufman, and Klaus Mueller are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-4400. E-mail: qfeng, fxu, fzhe, nneophyt, ari, mueller@cs.sunysb.edu.*

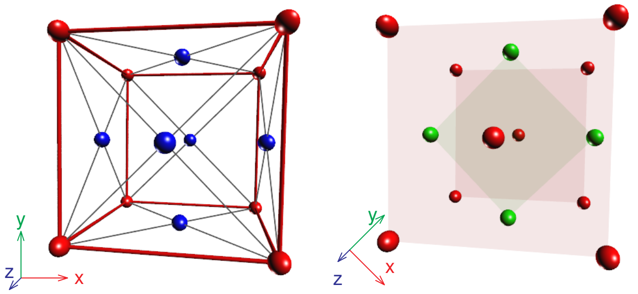
Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

2 FCC Lattice

An FCC lattice consists of simple CC cubic cells with additional sampling points located at the center of each cell face (see Fig. 1a). According to the lattice definition described in Section 1, any FCC lattice site can be constructed via linear combination of the three basis vectors $X = (1,0,0)$, $Y = (0,1,0)$ and $Z = (0.5, 0.5, \sqrt{2}/2)$, which can be obtained by defining an appropriate rotated coordinate system (see Fig. 1b). This definition scheme is adopted in our paper due to its simplicity, where the FCC lattice can be decomposed into two interleaved sub cubic lattices with a deviation vector of $v = (0.5, 0.5, \sqrt{2}/2)$. This representation provides a framework that enables quick indexing and efficient implementation of many basic lattice operations. For instance, the mapping from an arbitrary FCC lattice site of index (i, j, k) to its CC coordinates (x, y, z) can be defined by the following equations:

$$\begin{aligned} x &= i + (k \mathbf{\ mod\ } 2)/2, \\ y &= j + (k \mathbf{\ mod\ } 2)/2, \\ z &= \sqrt{2}k/2. \end{aligned} \tag{1}$$

Finding the nearest FCC lattice site to a sample point can be implemented by first looking for two neighbors having the shortest Euclidean distance to the sample point in the two sub cubic lattices using Eqn. 1, and then selecting the closer one between them.

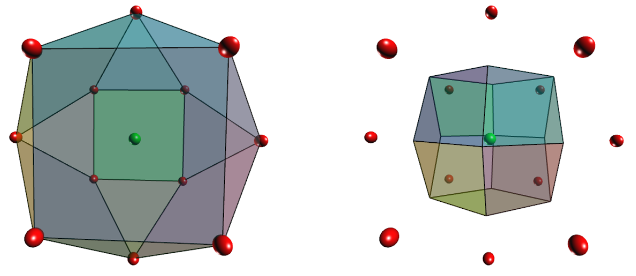


(a) (b)

Fig. 1. An FCC lattice can be constructed by (a) adding (blue) sites at the centers of the faces of cubic cells; (b) interleaving 2D square grids (red and green semi-transparent slices), where slice $2i+1$ is shifted from slice $2i$ by v. The second form (b) can be constructed by rotating the first one (a) $\frac{\pi}{4}$ about the z axis.

The geometric layout of the FCC lattice also gives rise to its higher angular discretization granularity than both CC and body-centered cubic (BCC) lattices, which is important for our rendering framework. Each site in the FCC lattice has direct links to a total of 12 nearest neighbors, in contrast to 8 and 6 in the BCC and CC lattices, respectively. This is the best angular discretization rate that any 3D regular lattice can achieve, since in R^3 the maximum number of sphere of radius 1 that can simultaneously touch the unit sphere (i.e., the kissing number) is 12 [6]. This unique feature has important implications for sampling and interpolation, as we will discuss further below. For example, when a particle is scattered at an FCC lattice site, it has 12 possible moving directions, which is 50% more than a BCC lattice and 100% more than a CC lattice. In addition, the 12 links of an FCC lattice site are symmetric under rotation and reflection, which supports a relatively simple computation framework. Fig. 2a shows the cuboctahedron defined by the 12 closest neighbors of a FCC site, while Fig. 2b shows the Voronoi cell of an FCC lattice site, which is essentially a rhombic dodecahedron.

Finally, the reciprocal lattice of FCC yields its representation in the frequency domain, which is essentially a BCC lattice [29]. As it will be described in the next section, this property gives rise to its near-optimal sampling behavior that is capable of reconstructing original signals with a minimum number of samples. It has been shown in the literature [29] that the FCC sampling scheme requires 13.4% and 23% fewer samples in R^2 and R^3 domain compared to CC lattices, respectively. Thus, the FCC lattice presents a much more efficient spacial sampling scheme over the traditional CC lattice.



(a) (b)

Fig. 2. (a) The 12 neighbors (red spheres) of an FCC lattice site (green sphere) forms a cuboctahedron; (b) The voronoi cell of an FCC lattice site (green sphere) is a rhombic dodecahedron.

3 Sampling on FCC

In any lattice used for volume visualization, the lattice sites are discretized or sampled from R^n, and an efficient, preferably optimal, sampling scheme is paramount. An optimal lattice structure captures information in the hyper-volume R^n using the least number of sampling points. Assuming an isotropic and band-limited sampling function, the resulting frequency support of a sampling point is a hyper-sphere, surrounded by a set of alias replicas. Hence, the most efficient sampling scheme arranges the replicated (hyper-spherical) frequency response as densely as possible in the frequency domain to avoid overlapping of the aliased spectra. As demonstrated in multi-dimensional signal theory [8] an optimal sampling scheme is obtained when the frequency response of the sampling lattice is an optimal sphere packing lattice [6]. Optimal sampling lattices can achieve up to 13.4%, 29.3%, and 50% of savings in 2, 3 and 4 dimensions, and they have been used in volume visualization [10, 22, 29] with high quality image results.

In three dimensions there are infinite optimal sphere packings including the FCC lattice and the HCP (hexagonal closed packing). The spatial equivalent of the FCC lattice in the frequency domain is the BCC lattice, which is the inverse Fourier transform of the FCC and vice-versa. The FCC lattice in the spatial domain corresponds to the BCC lattice in the frequency domain which is not an optimal sphere packing, and the FCC lattice achieves about 23% of savings over the CC lattice in terms of sampling efficiency. It was chosen for our global illumination framework because it is the lattice that maximizes uniform angular discretization with its kissing number of 12. The HCP is another candidate with an optimal kissing number of 12. In strict mathematical definition, HCP is not a lattice but can be defined as the union of the lattice L with generation vectors $(1,0,0)$, $(1/2, \sqrt{3}/2, 0)$ and $(0,0,\sqrt{8/3})$ and the translate $L + (1/2, 1/\sqrt{12}, \sqrt{2/3})$. However, it has a bias towards the z-direction. For any link with direction $d = (d_x, d_y, d_z)$ such that $d_z \neq 0$, the link of direction $-d$ does not exist. When a ray or photon moves from one of such links and is not absorbed or scattered at the lattice site, it cannot continue straight along its original direction. Therefore, HCP is not symmetric and thus unsuitable for use in our framework.

Given the initial assumption that the represented function is hyper-spherically band-limited, the ideal choice for the reconstruction function is also a radially symmetric kernel. We have studied a set of Gaussian reconstruction filters and have found that the relatively narrow Gaussian $f(r) = e^{-2r^2}$, offers good frequency behavior and reasonable overlap between neighboring sites. We have also tested this experimentally by reconstructing a constant cubic volume (where $f(x,y,z) = 1$) sampled on an FCC lattice and measuring the mean squared error for varying reconstruction kernel extents.

4 Rendering Methods

There are three major processes for the interaction between light and volumetric objects: emission, absorption and scattering. The differential equation [21] of light propagation in volumetric objects is

$$\begin{aligned}\frac{dI(x,\omega)}{ds} &= -\sigma_t(x,\omega)I(x,\omega)+\sigma_e(x,\omega)+\\ &\quad \sigma_s(x,\omega)\int_{4\pi} f(x,\omega,\omega')I(x,\omega')d\omega' \qquad (2)\end{aligned}$$

where $I(x,\omega)$ is the light intensity at position x in direction ω, σ_t is the *extinction* or *attenuation* coefficient (combination of absorption and out-scattering, $\sigma_t = \sigma_a + \sigma_s$), σ_s is the *scattering* coefficient or the scattering probability per unit distance. $f(x,\omega,\omega')$ is the *phase function* representing the conditional probability of a photon from direction ω to be scattered in direction ω' assuming that the photon is scattered, and it obeys $\int_{4\pi} f(x,\omega,\omega')d\omega = 4\pi$. In most naturally occurring media, σ_t and σ_s are independent of direction ω. Also, most media is *isotropic* and the phase functions only depend on the cosine of the angle θ between ω and ω'. The solution of Eqn. 2 is:

$$I(x,\omega) = \int_0^D R(x-s\omega,\omega)\exp\left(\int_0^D \sigma_t(x-t\omega)dt\right)ds, \qquad (3)$$

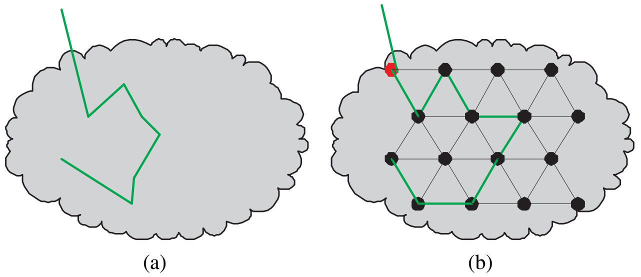
$$R(x,\omega) = \sigma_s(x,\omega)\int_{4\pi} f(x,\omega,\omega')I(x,\omega')d\omega' \qquad (4)$$

where D is the distance from x to the view point and $R(x,\omega)$ is the radiance at position x from direction ω.

We describe a new two-pass rendering algorithm on FCC lattices. In the first pass, photons are emitted from light sources and the photon energy is distributed in the scene, illuminating the media. In the second pass, a ray tracing method is used to generate the final image. At each sampling point x on the ray of direction $-\omega$, the radiance $R(x,\omega)$ is estimated by the photons in a small neighborhood of x for shading.

4.1 Diffuse Photon Tracing

For volumetric objects where the dominant effect is diffusion, we have developed a new algorithm to trace photons on the lattice links. The volumetric objects are sampled in a finite region of the FCC lattice. As shown in Fig. 3 on a 2D FCC lattice, when a photon is emitted from a light source, the nearest lattice site to the first hit point on the lattice boundary is calculated and the moving direction of the photon is discretized to one of the link directions. The photon will be traced on the links between lattice sites and its path is composed of line segments of lattice links.


(a)

(b)

Fig. 3. Illustration of tracing photons on a 2D FCC lattice. (a) The photon path in green color in the real medium. (b) The photon path on a 2D FCC lattice. The circles represent lattice sites and the black line segments represent lattice links. The red circle is the nearest lattice site to the photon intersection point with the medium boundary. The photon path is composed of the green lattice links.

A photon emitted from the light sources has an arbitrary direction ω. We first convert the photon direction stochastically to one of the lattice links. An FCC lattice site x_i has 12 closest neighboring sites of equal distance, which forms a cuboctahedron as shown in Fig. 2a. The ray originated from site x_i with direction ω intersects the cuboctahedron at point v on one of its 14 faces. Denoting the vertices of the face containing v as $v_0, v_1, \cdots, v_m$ $(m = 2,3)$ in counterclockwise order and letting $\omega_k = v_k - x_i$, the probability of the photon direction changing to ω_k is defined as the barycentric coordinates [11]:

$$p_k = \frac{w_k}{\sum_k w_k}, \quad w_k = \prod_{t\neq k-1,k} A(v,v_t,v_{t+1}) \qquad (5)$$

where $A(v,v_t,v_{t+1})$ is the area of triangle v, v_t, v_{t+1}. The barycentric coordinates of v are well defined with Eqn. 5 because all the faces of the cuboctahedron are regular polygons.

When arriving at a lattice site x_i from direction ω_j the photon might be absorbed, be scattered or continue moving along the extended link of ω_j. We use the absorption and the scattering coefficients, $\sigma_a(x_i,\omega_j)$ and $\sigma_s(x_i,\omega_j)$, to represent the probabilities of such events. (Please note that this approximation is only correct when the link length l is small compared to $1/\sigma_t$. The actual absorption probability should be $exp(-\int_0^l \sigma_a(x_i,\omega_j))ds$. Because the links on an FCC lattice are of identical length, we use σ_a, σ_s and σ_t here for convenience.) The Russian roulette technique [23] is used to determine whether the photon is absorbed, scattered or transmitted. In detail, the program generates a random number $\xi \in [0,1)$ and the photon:

$$\begin{cases}\text{is absorbed} & \text{if } \xi \in [0,\sigma_a(x_i,\omega_j)),\\ \text{is scattered} & \text{if } \xi \in [\sigma_a(x_i,\omega_j),\sigma_a(x_i,\omega_j)+\sigma_s(x_i,\omega_j)),\\ \text{is transmitted} & \text{if } \xi \in [\sigma_a(x_i,\omega_j)+\sigma_s(x_i,\omega_j),1).\end{cases} \qquad (6)$$

In contrast to the deterministic procedures of absorption and transmission, stochastic scattering requires further processing. In traditional methods, a phase function $f(x,\omega,\omega')$ is utilized to describe the probability of a photon being scattered at location x with an input direction ω and an output direction ω'. The computation of the new direction is performed using importance sampling. In the widely used models such as the Schlick model [1], the probability depends on $\cos(\theta) = \omega\cdot\omega'$ only, thus the importance sampling returns the value of $\cos(\theta)$. In order to compute ω', a local coordinate system at the scattering position has to be constructed to convert spherical angles to direction vectors, which is computationally expensive.

In our lattice illumination framework, the computation of the scattering process is greatly simplified because photons only move along discretized lattice links. Here, the continuous phase function $f(x,\omega,\omega')$ is discretized to $f(i,j,k)$, which represents the probability of a photon located at site x_i being scattered from the input link ω_j to the new output link ω_k. In practice, this discrete phase function is constructed as a 2D table of $n\times n$ resolution to represent all possible combinations of input/output links on a lattice site ($n = 12$ for FCC lattices). The generation of the discrete phase function via sampling and normalizing the continuous phase function is described by the following equation:

$$f(i,j,k) = \frac{f(x_i,\omega_j,\omega_k)}{\sum_{t=0}^{n-1} f(x_i,\omega_j,\omega_t)}. \qquad (7)$$

A more accurate method is to calculate the integral over the angular space Ω defined in Eqn. 8, which can be solved numerically for arbitrary continuous phase function:

$$\begin{aligned} f(i,j,k) &= \int_\Omega closest(\omega,\omega_k)f(x_i,\omega_j,\omega)d\omega \\ closest(\omega,\omega_k) &= \begin{cases}1 & \text{if the closest link to } \vec{d} \text{ is } \omega_k,\\ 0 & \text{otherwise.}\end{cases} \end{aligned} \qquad (8)$$

Importance sampling of discrete phase functions is simply a binary search for a given random number ξ such that $\sum_{t=0}^{k} f(i,\omega_j,\omega_t) \le \xi$ and $\sum_{t=0}^{k+1} f(i,\omega_j,\omega_t) > \xi$. The complexity of this binary search is

$O(\log(n))$, which is very efficient because n is usually very small ($n = 12$ for FCC lattices).

After determining the photon behavior of each encounter, we record their activity information on the lattice sites. Basically, a stored photon represents a possible light path from the light sources to its location. This information is used in the following rendering pass, where the irradiance of sampling positions is estimated with the stored light paths within a small neighborhood. The lattice-based framework enables us to store photons in a 3D array and the position of the photons is implicitly defined by their index in the array. Moreover, the quantized directions are encoded in a byte using the link index. Since the link vectors are known a-priori, we adopt an optimized solution for photon direction storage, where photons with the same direction are grouped together. Here, a 1D array $E(\omega_j)$ of n elements is used for a lattice site x_i, such that $E(\omega_j)$ is the total energy of the photons at x_i with direction ω_j. Due to the employed Russian roulette technique [23], the photon energy does not change until it is absorbed. Therefore, the stored photons all have the same energy and only the integral number of photons need to be stored at link (i, j). Given the maximum possible number of photons stored on the links, the unsigned byte or short format can be used to represent the actual photon counts instead of storing individual floating point energy values. In other words, we store all photons in a 4D integer array $E(i, \omega_j)$ with three dimensions of site position and one dimension for link direction.

After the recording of diffuse photons, rays are traced from the image plane into the FCC lattice to collect irradiance values. At each sampling point x, the radiance is estimated by the photons inside a small spherical region centered at x. With our 4D array of photon counts, the radiance in Eqn. 4 is calculated with the following simplified formula:

$$R(x,\omega) = \sum_X \sum_j \sigma_s(x,\omega_j) f(x,\omega,\omega_j) g(x' - x) E(x', \omega_j) dx' \qquad (9)$$

where ω is the reverse ray direction, f is the continuous phase function and X is the set of lattice sites in the search region. g is a normalized smoothing filtering function used for removing high-frequency noise. Because the lattice sites are positioned regularly, searching for the lattice sites in a sphere is simple and efficient. In our experiments, the medium is isotropic and the phase function only depends on the angle between the ray direction and the lattice link ω_j. The dot product of ω and ω_j can be pre-computed and reused for all the sampling points on one ray in the rendering process, which yields an efficient implementation of the radiance estimation.

Our new algorithm greatly simplifies the computation of photon-volume interaction and photon storage. Therefore, our program can trace millions of photons in a short time, which is good for improving image quality by removing the stochastic noise and variance without excessive smoothing in the radiance estimation. Moreover, our method is general and can use arbitrary phase functions including those of strong backward scattering.

4.2 Specular Photon Tracing

The method described in Section 4.1 is capable of calculating multiple scattering events for participating media and volumes where diffusion is the dominant effect. However, specular reflection and refraction may exhibit ray effects when discretized rays hit smooth specular surfaces. This effect has been discovered by the radiative heat transfer community [4] and found to be caused by the discretization of scattering directions when accurate directions are needed for specularity.

To mitigate this ray effect, we describe an enhanced algorithm called specular photon tracing, where every photon is associated with its accurate direction ω and start position x. In each time step, the photon moves on the FCC lattice and the new sampling position is calculated by $x = d \times \omega$ where d is the step size. The lighting properties σ_a and σ_s are sampled to decide whether the photon is absorbed, scattered or transmitted at x (Eqn. 6). If the photon is scattered, the new direction ω is computed with the continuous phase function. The Russian roulette technique is again used to avoid photon energy change.

The O-Buffer data structure [25] is used to store the photon information compactly, where each lattice site stores a sequence of photons in its Voronoi cell (Fig. 2b). For each stored photon at position x, the nearest lattice site x_i is computed and only the offset o from x to x_i is stored. The offset o is quantized into 256 levels in each axis so that o can be compactly represented in 3 bytes. This 3-byte representation increases the photon position accuracy by 256 times of the lattice resolution, while it only needs 25% of the storage space of the traditional floating point representation. Because the search radius for the radiance estimation is usually much larger than the link length, it is good enough for most rendering applications. The Voronoi cell (Fig. 2b) of an FCC lattice site is a rhombic dodecahedron. Assuming a unit distance between neighboring sites, the distance from the lattice site to the vertices of its Voronoi cell is $\frac{\sqrt{2}}{2}$. The maximum error introduced by this offset quantization scheme is $\frac{\sqrt{3}}{2} \cdot \frac{1}{255} \cdot \frac{\sqrt{2}}{2}$.

For photon direction encoding, vectors ω are converted to spherical coordinates and represented with 2 bytes [15]. Because the photon energy does not change, only one byte is used to record the color channel of the photon. In total, the storage space of one photon is just 6 bytes.

In the rendering pass, the radiance at each sampling point is estimated with the photons stored in the lattice photon O-Buffer. For a spherical search neighborhood S with radius r, the lattice sites in the sphere S' of radius $r + \frac{\sqrt{2}}{2}$ are retrieved because the maximum distance from a lattice site to the photons stored in it is $\frac{\sqrt{2}}{2}$. Then, the photons stored in these lattice sites are visited. If the distance to the sampling point x is larger than r, the photon is discarded. The radiance at x is summed over all the photons inside S with

$$R(x,\omega) = E \sum_p \sigma_s(\omega_p, x) f(x,\omega,\omega_p) g(x_p - x) \qquad (10)$$

where x_p and ω_p are the photon position and direction, respectively. The term E is the energy of the photon, which is the same for all photons since the Russian Roulette scheme is used.

Our diffuse photon tracing algorithm is capable of tracing multi-million photons in seconds. With the FCC lattice, the photons move on the lattice links and are scattered only on the lattice sites. Therefore, the most time consuming steps in traditional methods such as sampling the lighting properties, calculating scattered directions with phase functions are greatly simplified. Its major disadvantage is the ray effect, which cannot be neglected in certain cases, for example, in reflection, refraction or scattering on specular surfaces, and hard shadows. The specular photon tracing solves this problem but is much slower.

5 Implementation

We have implemented the algorithms described in Section 4 to render participating media and volumetric datasets. Since the reconstruction process in the current scanning modalities such as MRI and CT only produce rectilinear data, the FCC data we used are generated by resampling existing rectilinear volumes or voxelizing geometric objects. In the resampling process, a windowed sinc filter has been used to obtain optimal signal reconstruction quality.

We use the incremental triangle voxelization method [7] to voxelize polygonal models to the FCC lattice. The original method was proposed for volumes of CC lattices, but the employed distance-based method enables its direct application to the FCC lattice.

For surface voxelization, the volume density of a lattice site p is determined by the distance between p and its closest triangle. Each triangle is processed in the following manner. For each lattice site p in the neighborhood of the triangle, the distance d between p and the triangle is computed. The distance d is positive if p is in the normal direction of the triangle, or negative if p is in the reverse direction. If $|d|$ is smaller than the previously stored absolute value of the distance, $|d|$ replaces the previous stored value and the density of p is updated with the following equation:

$$density = \begin{cases} 1 & \text{if } d < -W, \\ 0 & \text{if } d > W, \\ 0.5 \times (1 - \frac{d}{W}) & \text{otherwise,} \end{cases} \tag{11}$$

where W is the width of the oriented box filter. While for a CC lattice the optimized width was estimated to be $2\sqrt{3}$ voxel units [7], we estimate that for an FCC lattice, the optimized width of the filter is decreased to 2 lattice units. Based on the surface voxelization result, the interior of the solid is voxelized using seed growing.

For inhomogeneous participating media, such as clouds and smoke, the absorption coefficient $\sigma_a(x)$ and scattering coefficient $\sigma_s(x)$ typically do not rely on the light direction, although our algorithm is capable of rendering anisotropic media. The volume data of the medium usually defines the density field $\rho(x)$ of particles. We assume that the $\sigma_a(x)$ and $\sigma_s(x)$ are proportional to the local density $\rho(x)$. For FCC lattices where the lattice links have unit lengths, it implies:

$$\sigma_a(x, \omega_j) = \sigma_a \rho(x), \text{ and } \sigma_s(x, \omega_j) = \sigma_s \rho(x). \tag{12}$$

where σ_a and σ_s are user-defined scaling coefficients. In real world phenomena, most practical participating media are isotropic and the phase function $f(x, \omega, \omega')$ does not vary upon position x. In our implementation, $f(x, \omega, \omega') = f(\omega \cdot \omega')$ is described with the Schlick model [1]. The participating medium is represented with an FCC lattice of densities and all the coefficients are calculated by scaling $\rho(x)$.

Our algorithms support chromatic participating media, where the coefficients $\sigma_a(x, \lambda)$ and $\sigma_s(x, \lambda)$ are wavelength-dependent. We implement it by defining scaling factors of absorption and scattering in RGB channels. The photons emitted from the light sources can be red, green or blue randomly. The photon tracing program calculates $\sigma_a(x, \lambda)$ and $\sigma_s(x, \lambda)$ with proper scaling factors in the color channel of the traced photons. In the rendering pass, the opacity value α of a sampling point is the average extinction of those in three channels:

$$\alpha = \frac{1}{3}(\sigma_t(x, \lambda_r)^d + \sigma_t(x, \lambda_g)^d + \sigma_t(x, \lambda_b)^d) \tag{13}$$

where d is the step size.

For general volume datasets, transfer functions have been exploited to define lighting properties from the density field. Some 2D transfer functions might also use the gradient information. Our framework is general and capable of integrating any transfer function as long as the input data (density, gradient or any other data) of the transfer function is defined on the lattice. In our current implementation, a 1D transfer function is defined for σ_a and σ_s in each RGB channel. Eqn. 13 can be used to compute the opacity α or a separate transfer function can be defined for α.

Following the photon tracing computation, a single-pass ray-casting approach is employed on the GPU to render the diffuse photon tracing results. We used an algorithm similar to [19], except that the sampled volume density is used for obtaining the scattered coefficient as well as transparency values to composite the final radiance results. Here, in addition to the density volume, an additional photon storage volume that records the diffuse photon distribution on each lattice site is sampled to composite ray values. Both the density and the diffuse photon volumes are FCC lattices, stored and indexed as described in Section 2. The diffuse photon storage table is essentially a 12-element array, each of which records the number of photons stored along 12 different lattice links. A two-byte unsigned short is allocated for each such node to provide a count of up to 65536. To compute the radiance estimation at each sampling point, dot-products of each lattice link with the current viewing ray are used to weight individual diffuse radiance value, which is given by indexing the photon distribution previously computed on the lattice. The 12 weighted values are then summed up to yield the final contribution. Filtering on all lattices uses the Gaussian kernel of size 2 and the sum is normalized at the end.

6 Results

We have implemented our algorithms on a 3.4GHz PC with 3GB memory and a Geforce 8800 GTX graphics card. All the resulting images are of 512×512 resolution and cropped in Figs. 4-8.

Fig. 4 demonstrates the rendering results of participating media with our algorithms. We use a single light of white color placed on top of the smoke dataset. Table 1 gives the times of different algorithms used to render the corresponding images.

In Fig. 4b, the eccentricity coefficient k of the Schlick phase function is set to 0.2. The absorption and scattering coefficients are $\sigma_a = 0.08$ and $\sigma_s = 0.2$, respectively. Fig. 4c uses the same coefficients except that the eccentricity k is -0.5 for strongly backward scattering. The photon tracing of 1 million photons and subsequent rendering passes cost about 11 and 48 seconds on the CPU, respectively. With GPU acceleration, the time of the rendering stage is reduced to 2.4s. This performance is significantly faster than the original photon mapping method [16], where the tracing 10K photons in a cloud model of similar size takes 8 seconds on an HP computer of 16 180MHz PA-8000 processors, while rendering a 1024 pixel wide image takes 92 seconds. Note that although a higher resolution is used in Jensen and Christensen's method [16], the complexity of the algorithms is mainly determined by the number of photons, of which our example generates 100 times more. A major further enhancement of our implementation can be achieved by incorporating empty space skipping or adaptive sampling techniques that are used by Jensen and Christensen [16].

Fig. 4d has been rendered using specular photon tracing with the same medium properties as Fig. 4b, and the rendering time is similar to traditional photon mapping methods [16]. With our compact FCC O-Buffer data structure, 5.8 million stored photons only consume 35MB of memory space. The search radius for radiance estimation is changed from 2.0 to 3.0 to remove the stochastic variance. Given the same number of photons, diffuse photon tracing is approximately 21 times faster than specular photon tracing and the corresponding rendering pass is 15 times faster. We observed that the image produced from our lattice-based framework using diffuse photon tracing (Fig. 4b) is comparable to the image using specular photon tracing (Fig. 4d), and has a similar quality and appearance of those computed with traditional photon mapping methods (such as Figure 12.10 of [23]). However, our framework has much better performance.

In Fig. 4e, the absorption coefficient has been made wavelength-dependent and the values in RGB channels are $\sigma_a = (0.08, 0.15, 0.3)$, while the scattering coefficient is the same as in Fig. 4b. The time of the rendering pass increases to 52.7 seconds mainly because the radiance estimation is performed in 3 channels. In Fig. 4f, the scattering coefficient in the blue channel has been changed to 0.4 and the eccentricity coefficient k has been changed to 0.5. In Figs. 4e and 4f, the upper part of the smoke is grey but the lower part under the shadow of the upper part becomes orange because of the different absorption and scattering coefficients in the RGB channels.

Fig. 5 demonstrates another example of participating media. The eccentricity coefficient k of the Schlick phase function is 0.2. The absorption and scattering coefficients are $\sigma_a = 0.05$ and $\sigma_s = 0.1$, respectively. One million photons have been traced in 12.7 seconds and the rendering pass amounted to 98.5 seconds on the CPU and 5.5 seconds on the GPU.

Fig. 6 displays the foot of the visible human CT data. Fig. 6a is rendered using a ray casting method with local illumination. Fig . 6b has been rendered using our diffuse photon tracing algorithm. The bone appears semi-translucent and brighter and has soft self-shadows due to multiple scattering. In Fig. 6c, the muscle and soft tissue are displayed with red color with the absorption coefficient similar to 6a.

Fig. 7 and Fig. 8 are the rendering results of the engine and lobster data, respectively. Fig. 7a has been rendered using ray casting with local illumination and the other two images have been rendered with our framework and indeed the objects appear substantially different. In Fig. 7b, the material absorbs green and blue photons faster and becomes more red gradually through the light direction. In Fig. 7c, the high density region appears saturated red for emphasis and the sur-

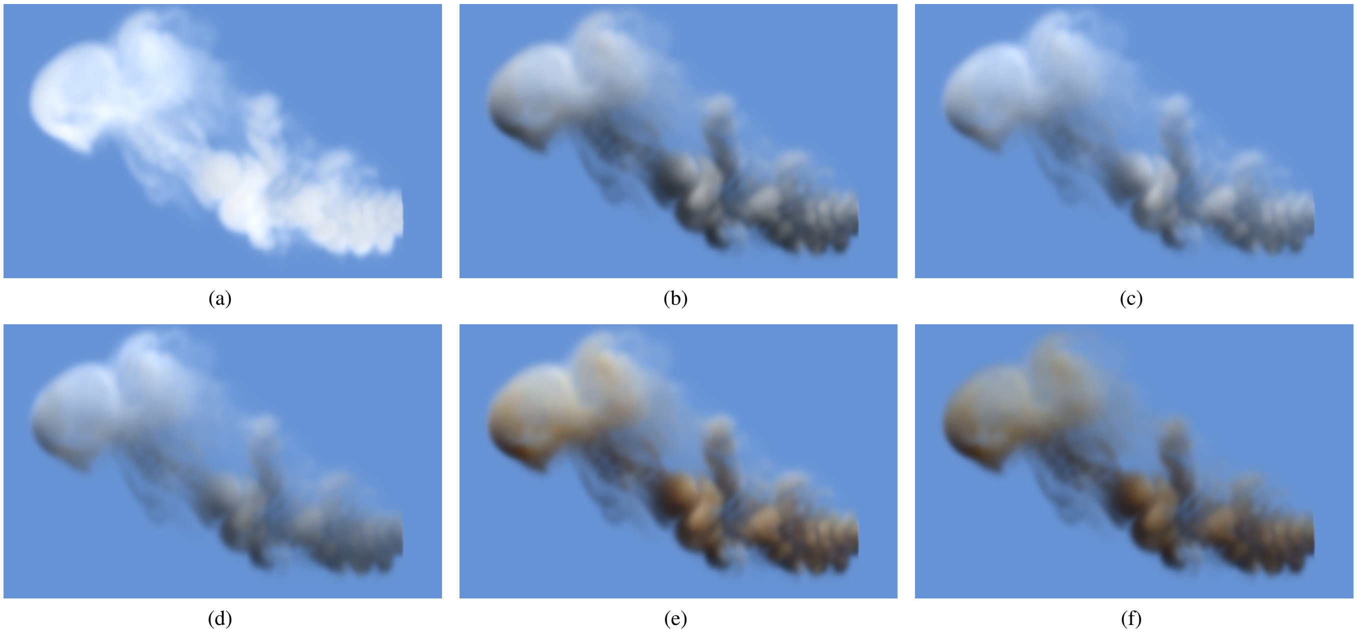

(a) (b) (c) (d) (e) (f)

Fig. 4. Inhomogeneous smoke rendered with global illumination (multiple scattering) and an anisotropic phase function. The original data is $100 \times 100 \times 40$ and we sampled it into a $108 \times 100 \times 56$ FCC lattice with a windowed sinc filter. The algorithms used are (a) ray casting; (b) our diffuse photon tracing; (c) our diffuse photon tracing with strong backward scattering; (d) our specular photon tracing; (e) and (f) our multi-channel diffuse photon tracing.

Table 1. Times used to render the smoke in Fig. 4. From left to right, the columns represent the algorithm used, the number of photon traced (in millions), photon tracing time (in seconds), and rendering time (in seconds) on the CPU and the GPU.

	Algorithm	**Photon count (M)**	**Photon tracing (s)**	**CPU rendering (s)**	**GPU rendering (s)**
Fig. 4a	ray casting	N/A	N/A	20.6	1.0
Fig. 4b	diffuse photon tracing	1.0	11.3	48.1	2.4
Fig. 4c	diffuse photon tracing	1.0	11.7	48.1	2.4
Fig. 4d	specular photon tracing	0.1	27.0	729.0	N/A
Fig. 4e	multi-channel diffuse photon tracing	3.0	48.5	52.7	2.6
Fig. 4f	multi-channel diffuse photon tracing	3.0	48.6	52.7	2.6

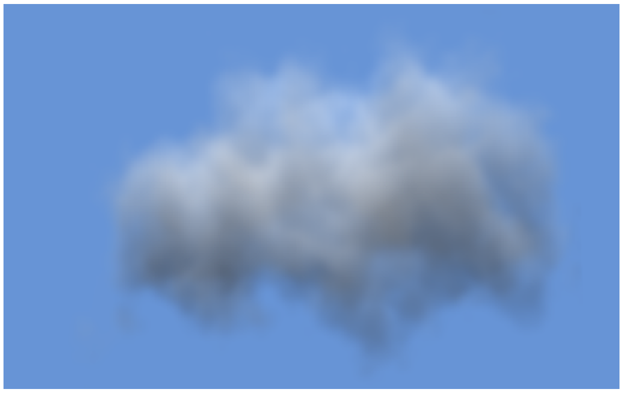


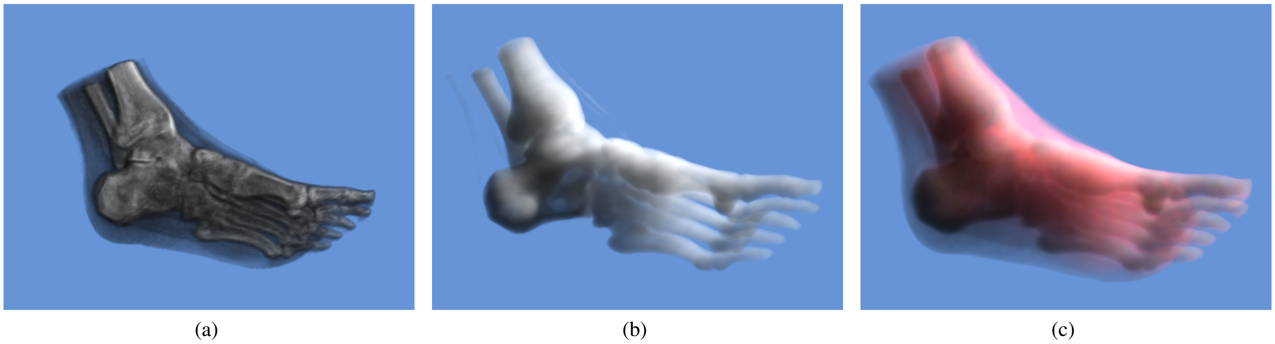
Fig. 5. Cloud rendered with our diffuse photon tracing. The resolution of the data is $96 \times 74 \times 143$.

rounding region is less saturated red due to color bleeding. The material of the objects is easily controlled and changed using user specified transfer functions. In Fig. 8b, the lobster shell absorbs green and blue photons faster than red ones and scatters red photons more than green and blue ones. The shell appears red and the muscle casts soft shadows onto itself. Fig. 8c displays the data from a different camera position.

7 Discussion

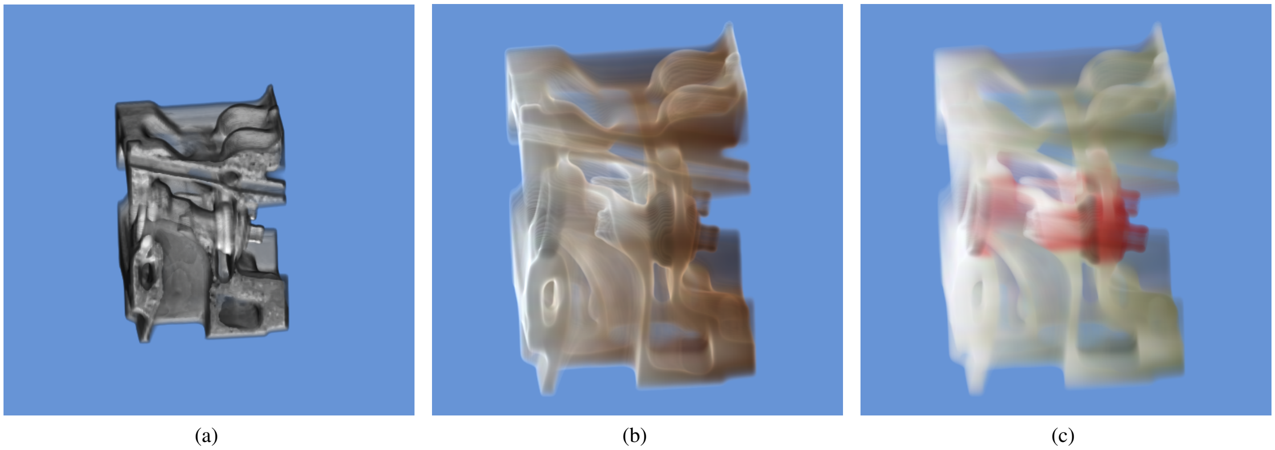
In our framework, a photon is saved at every step of the first pass, regardless of whether it is scattered or not. A stored photon represents a possible path from the light sources. In the ray tracing pass, the radiance estimation actually calculates the density of photon paths at the sampling positions. In contrast, the traditional photon mapping method uses the probabilistic sampling technique to calculate the step size, and the expected step size is $1/\sigma_t$. Usually σ_t is small and tracing a photon can generate many more stored photons in our algorithms than in photon mapping. Moreover, the FCC lattice provides a more efficient data structure to store photons. In diffuse photon tracing, each lattice site stores multiple photons. In radiance estimation, the contribution of multiple photons on a lattice link is computed jointly using Eqn. 9. Suppose the radiance estimation searches photons in the spherical neighborhood S, and there are k_0 sites and k_1 photons inside S. It costs $O(k_0)$ time for diffuse photon tracing and $O(k_0 + k_1)$ for specular photon tracing. However, it costs $O(k_1 + \log n)$ time with the k-d tree data structure in photon mapping, where n is the total number of photons. Also, the k-d tree need to be built before the rendering pass, which costs $O(n \log n)$ time. However, our algorithms do not need such a preprocessing step.

There are other simplified lighting models for participating media and volumes [13, 14, 18, 26]. For all these methods, only forward scattering is considered and lighting values are propagated from slice to slice. The value of each pixel is calculated by sampling and attenuating neighboring pixels (up to 4) on the previous slice. In other words, forward scattering is calculated in several directions in the 2π



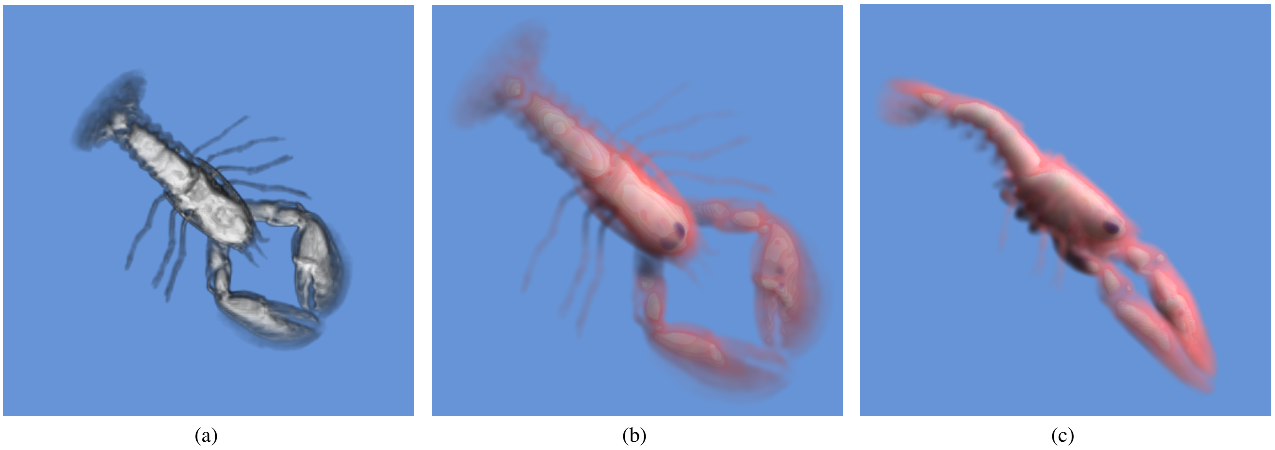
(a) (b) (c)

Fig. 6. Global illumination of a CT scan of the visible human foot. The original data is 128^3 and the sampled FCC lattice is $128 \times 128 \times 180$. The images are rendered with (a) ray casting with local illumination; (b) our diffuse photon tracing; (c) our multi-channel diffuse photon tracing.



(a) (b) (c)

Fig. 7. Global illumination of an industrial CT scan of an engine. The original data is $128 \times 128 \times 64$ and the sampled FCC lattice is $136 \times 136 \times 98$. The images are rendered with (a) ray casting with local illumination, (b) and (c) our multi-channel diffuse photon tracing.



(a) (b) (c)

Fig. 8. Global illumination of a CT scan of a lobster. The original data is $128 \times 127 \times 28$ and the sampled FCC lattice is $114 \times 113 \times 35$. The images are rendered with (a) ray casting with local illumination; (b) and (c) our multi-channel diffuse photon tracing.

Table 2. Rendering of the foot, engine and lobster data in Figs. 6, 7 and 8. The meaning of the columns are the same as in Table 1.

	Algorithm	Photon count (M)	Photon tracing (s)	CPU rendering (s)	GPU rendering (s)
Fig. 6b	diffuse photon tracing	1.0	27.0	98.0	4.7
Fig. 6c	multi-channel diffuse photon tracing	3.0	86.1	118.6	5.8
Fig. 7b	multi-channel diffuse photon tracing	9.0	136.4	110.6	5.4
Fig. 7c	multi-channel diffuse photon tracing	9.0	147.9	110.7	5.4
Fig. 8b	multi-channel diffuse photon tracing	3.0	26.9	40.7	2.0
Fig. 8c	multi-channel diffuse photon tracing	6.0	53.9	37.9	1.9

solid angle. Our algorithms can handle scattering within the entire 4π solid angle. Also, our methods can store photons from multiple light sources in one volume, while previous methods do not support multiple light sources. The method of Max [20, 21] is more accurate than ours but runs slowly.

The idea of tracing photons on the lattice links is general and might be applied to other lattices such as the CC lattice. However, a CC lattice site only has 6 nearest neighbors, which is definitely not enough for discretizing the phase function. Consider a strongly forward scattering phase function, a photon arriving from a link can only move forward on its original direction or be scattered to 4 directions perpendicular to its incoming direction. Although we can add links between the secondary or tertiary neighbors, this solution needs to calculate and store the absorption and scattering probabilities of links with varying length (1, $\sqrt{2}$ and $\sqrt{3}$), thus making the photon tracing algorithm more complicated and time consuming. Instead, a FCC lattice site has 12 nearest neighbors, which is sufficient in photon tracing (in previous systems, only 4 or 5 directions are used for forward scattering). The link length in the FCC construction is uniform, which greatly simplifies the computation and storage: the absorption and scattering probabilities on 12 links of a site are the same and stored only once on the site. Moreover, the FCC lattice has better sampling efficiency than the CC lattice. With the same number of sites, FCC captures 23% more information. In addition, the maximum distance of an arbitrary point to its nearest site is $\frac{\sqrt{2}}{2}$ in FCC instead of $\frac{\sqrt{3}}{2}$ in CC, which means 18.4% less quantization error of photon positions with the O-Buffer data structure.

8 Conclusions and Ongoing Work

We have presented a novel framework for volumetric global illumination based on FCC lattices. Benefitting from the FCC lattice unique geometric and sampling properties, we were able to develop algorithms that can render participating media and volumes with multiple scattering effects. Our diffuse photon tracing algorithm renders high quality images at a speed significantly faster than conventional methods. We have also described a memory efficient specular photon tracing algorithm. In future work, our plan is to extend our framework to render hybrid scenes of volumetric datasets and surface objects with specular reflection and refraction.

Our current implementation only uses 12 links to the nearest sites on the FCC lattice. For future efforts, we consider incorporating the links connecting the secondary and tertiary neighbors to increase the angular discretization granularity. With these additional links, the phase function can be discretized with even higher precision, thus more accurate radiance estimation will be obtained. In doing so, a total of 42 neighbors with distance less than $\sqrt{3}$ will have to be considered and different link lengths will participate in the computation of the absorption and scattering coefficients. Hence, a more efficient data structure will be required for diffuse photon storage. We also consider applying our general framework to other types of particles or waves, such as phonons and Huygens' wavelets, to solve their energy and wave propagation problems. Last, we would like to accelerate the photon tracing pass on the GPU to improve the overall framework performance.

Acknowledgements

This work was supported, in part, by NSF CCF-0702699, NSF CAREER grant ACI-0093157, NIH grant CA082402, and NIH grant 5R21EB004099-02. The smoke data is courtesy of Dunc Nguyen and Ron Fedkiw. The engine data is courtesy of GE. The visible human data is courtesy of National Library of Medicine.

References

[1] P. Blasi, B. L. Saëc, and C. Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum*, 12(3):201–210, 1993.

[2] J. F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. *SIGGRAPH*, pages 21–29, 1982.

[3] E. Cerezo, F. Pérez, X. Pueyo, F. Seron, and F. Sillion. A survey on participating media rendering techniques. *The Visual Computer*, 21(5):303–328, June 2005.

[4] J. C. Chai, H. S. Lee, and S. V. Patankar. Ray effect and false scattering in the discrete ordinates method. *Numerical Heat Transfer Part B*, 24:373–389, 1993.

[5] S. Chandrasekhar. *Radiative Transfer*. Dover Publications, 1960.

[6] J. H. Conway, N. J. A. Sloane, and E. Bannai. *Sphere-packings, Lattices, and Groups*. Springer-Verlag, New York, NY, USA, 1987.

[7] F. Dachille and A. Kaufman. Incremental triangle voxelization. *Graphics Interface*, pages 205–212, May 2000.

[8] D. E. Dudgeon and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Prentice Hall Professional Technical Reference, 1990.

[9] D. S. Dummit and R. M. Foote. *Abstract Algebra*. John Wiley and Sons, second edition, 1999.

[10] A. Entezari, R. Dyer, and T. Moller. Linear and cubic box splines for the body centered cubic lattice. *Proceedings of IEEE Visualization*, pages 11–18, 2004.

[11] M. S. Floater, K. Hormann, and G. Kós. A general construction of barycentric coordinates over convex polygons. *Advances in Computational Mathematics*, 24(1–4):311–331, Jan. 2006.

[12] R. Geist, K. Rasche, J. Westall, and R. J. Schalkoff. Lattice-boltzmann lighting. *Proceedings of the Eurographics Workshop on Rendering Techniques*, pages 355–362, 2004.

[13] M. J. Harris and A. Lastra. Real-time cloud rendering. *Computer Graphics Forum*, 20(3):76–84, 2001.

[14] K. Hegeman, M. Ashikhmin, and S. Premože. A lighting model for general participating media. *Proceedings of Symposium on Interactive 3D Graphics and Games*, pages 117–124, 2005.

[15] H. W. Jensen. Global illumination using photon maps. *Proceedings of Eurographics Workshop on Rendering Techniques*, pages 21–30, 1996.

[16] H. W. Jensen and P. H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. *SIGGRAPH*, pages 311–320, 1998.

[17] J. T. Kajiya and B. P. V. Herzen. Ray tracing volume densities. *SIGGRAPH*, pages 165–174, 1984.

[18] J. Kniss, S. Premoze, C. Hansen, P. Shirley, and A. McPherson. A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics*, 9(2):150–162, April-June 2003.

[19] J. Kruger and R. Westermann. Acceleration techniques for GPU-based volume rendering. *Proceedings of IEEE Visualization*, pages 287–292, 2003.

[20] N. L. Max. Efficient light propagation for multiple anisotropic volume scattering. *Eurographics Workshop on Rendering*, pages 87–104, 1994.

[21] N. L. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

[22] N. Neophytou and K. Mueller. Space-time points: 4D splatting on efficient grids. *Proceedings of Symposium on Volume Visualization and Graphics*, pages 97–106, 2002.

[23] M. Pharr and G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[24] W. Qiao, D. S. Ebert, A. Entezari, M. Korkusinski, and G. Klimeck. VolQD: Direct volume rendering of multi-million atom quantum dot simulations. *Proceedings of IEEE Visualization*, pages 319–326, 2005.

[25] H. Qu and A. Kaufman. O-buffer: A framework for sample-based graphics. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):410–421, July-August 2004.

[26] K. Riley, D. S. Ebert, M. Kraus, J. Tessendorf, and C. D. Hansen. Efficient rendering of atmospheric phenomena. *Proceedings of Eurographics Symposium on Rendering*, pages 374–386, 2004.

[27] H. E. Rushmeier and K. E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. *SIGGRAPH*, pages 293–302, 1987.

[28] L. M. Sobierajski and A. E. Kaufman. Volumetric ray tracing. *Symposium on Volume visualization*, pages 11–18, 1994.

[29] T. Theußl, T. Möller, and M. E. Gröller. Optimal regular volume sampling. *Proceedings of IEEE Visualization*, pages 91–98, 2001.