Fast Computation of Single Scattering in Participating Media with Refractive Boundaries using Frequency Analysis

Yulin Liang, Beibei Wang, Lu Wang, Nicolas Holzschuch

Fast Computation of Single Scattering in Participating Media with Refractive Boundaries using Frequency Analysis

Yulin Liang, Beibei Wang†, Lu Wang*, and Nicolas Holzschuch

Abstract—Many materials combine a refractive boundary and a participating media on the interior. If the material has a low opacity, single scattering effects dominate in its appearance. Refraction at the boundary concentrates the incoming light, resulting in an important phenomenon called volume caustics. This phenomenon is hard to simulate. Previous methods used point-based light transport, but attributed point samples inefficiently, resulting in long computation time. In this paper, we use frequency analysis of light transport to allocate point samples efficiently. Our method works in two steps: in the first step, we compute volume samples along with their covariance matrices, encoding the illumination frequency content in a compact way. In the rendering step, we use the covariance matrices to compute the kernel size for each volume sample: small kernel for high-frequency single scattering, large kernel for lower frequencies. Our algorithm computes volume caustics with fewer volume samples, with no loss of quality. Our method is both faster and uses less memory than the original method. It is roughly twice as fast and uses one fifth of the memory. The extra cost of computing covariance matrices for frequency information is negligible.

1 Introduction

In translucent participating media with refractive boundaries, light gets refracted on the boundary into the volume, scatters only a few times and leaves the volume. The refractive boundary concentrates the incoming light in small parts of the material, resulting in high frequency effects called *volume caustics*. If the phase function is anisotropic, it can increase the effect by emphasizing the directional distribution of light. In this paper, we focus on single scattering effects, the main cause for volume caustics. Single scattering corresponds to light entering the material, being refracted at the first interface, scattering once inside the material, and leaving the material, being refracted a second time at the interface before reaching the camera. The presence of two refractions makes it difficult to compute single scattering using standard methods.

Walter et al. [1] and Holzschuch [2] introduced methods for accurate computation of single scattering effects in participating media with refractive boundaries. These methods are accurate, but also very expensive and highly dependent on scene complexity (triangle count). Point based methods by Wang et al. [3] are faster and solve for both high-order scattering (more than two scattering events) and low-order scattering, including single scattering. They work in two steps: in a preprocessing step, they distribute volume samples, caching both the geometry and illumination information, and organize them into a spatial hierarchy. During the rendering step, each camera ray is refracted into the volume and gets sampled into camera samples. Each camera sample accumulates the contributions from the volume samples and multiplies them with medium attenuation along the camera ray to obtain the final radiance of the camera ray. The point-based methods provide faster results for single scattering than the dedicated methods [1], [2], at the expense of a larger memory footprint.

- *Y. Liang is with School of Software, Shandong University.*
- *B. Wang is with School of Computer Science and Engineering, Nanjing University of Science and Technology. †Joint first author.*
- *L. Wang is with School of Software, Shandong University. *Corresponding author. E-mail: luwang_hcivr@sdu.edu.cn*
- *N. Holzschuch is with Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.*

In this paper, we introduce tools for frequency analysis of light transport [4] into the point-based illumination method by Wang et al. [3]. Specifically, we use covariance tracing, by Belcour et al. [5], [6], to compute the local frequency content for single scattering effects, using the covariance matrix of the frequency spectrum of the local light field. We adapt the original point-based algorithm in two ways: during the preprocessing step, we compute the covariance matrix for each volume sample, using the transport operators designed by Belcour et al. [6]. During the rendering step, we use the covariance matrix to predict the kernel size for radiance gathering at each volume sample. Adapting the kernel size to frequency information allows us to provide sharp volume caustics with much fewer volume samples, greatly reducing both the memory footprint and the cost of the preprocessing step. Our method provides pictures with better quality and smaller computation time compared to Wang et al. [3].

We review previous work on frequency analysis of light transport and on single scattering simulation methods in the next section. We then present covariance tracing [5] and the original point based method for participating media [3] in Section 3. We describe our algorithm in Section 4. In Section 5, we compare our method with previous works and reference solutions. We conclude in Section 6.

2 Related works

Accurate Single Scattering Walter et al. [1] introduced a method for accurate computation of single scattering effects in participating media with refractive interfaces. Their method computed the entry point for the ray connecting to the light source using Newton-Raphson optimization. Holzschuch [2] improved both accuracy and speed by computing the extent of the influence of each triangle over the camera ray. Computation time for his method depends strongly on scene complexity. In comparison, our method is faster, scales better and provides visually identical results.

Point-Based Light Transport Christensen [7] introduced Point-Based Global Illumination as a way to compute diffuse light transport efficiently. Direct illumination is computed and encoded using a mesh-less hierarchy of point samples. In a second step,

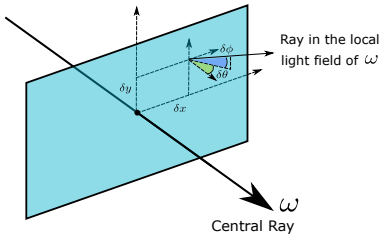


Fig. 1. The Local Light Field is defined as a 4D function around the center ray (ω), parameterized by two spatial coordinates (δ_x and δ_y) and two angular coordinates (δ_θ and δ_ϕ).

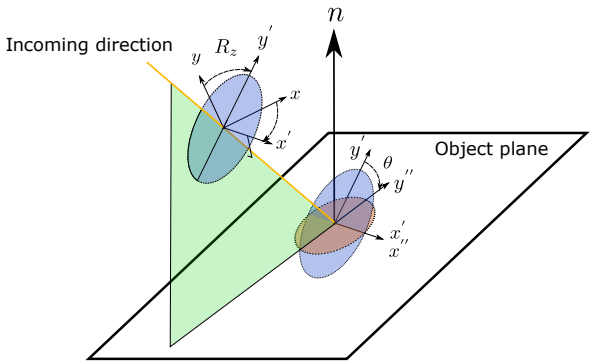


Fig. 2. Projection of the local light field to the local frame of the object.

they used local Z-buffers to compute illumination from the point samples, giving indirect illumination. Arbree et al. [8] extended the approach for subsurface scattering. Wang et al. [3] used point-based global illumination for participating media. They place volume samples inside the media, combined with the usual surface samples. Our method builds on theirs; introducing frequency analysis of light transport allows us to reduce the number of volume samples, decreasing the memory footprint and computation time.

Ray / Path Differentials Igehy [9] introduced *ray differentials* to ray tracing: using differential geometry to compute the footprint of a ray, and filter textures accordingly. This first paper only focused on specular paths. Suykens and Willems [10] extended the approach to arbitrary general paths, supporting glossy materials with a computation time scaling quadratically with path length and also neglecting the effects of the light source.

Frequency Analysis Durand et al. [4] introduced a framework for the frequency analysis of light transport. They express basic operations such as transport, occlusion or reflection as operators on the Fourier spectrum of the light field, allowing predictions on the frequency content of radiance at any point of the scene. Soler et al. [11] extended the approach, predicting image bandwidth and per-pixel variance for the incoming light to adaptively sampling primary rays. Egan et al. [12] extended the approach to compute motion blur for moving scenes, and reduce aliasing effects in sampling. Belcour et al. [5] introduced the covariance matrix to represent the Fourier spectrum of illumination in a compact way. The covariance matrix is associated to paths inside the scene, and computed using path tracing. Belcour et al. [6] extended the approach to participating media, and Belcour et al. [13] used it for anti-aliasing for bidirectional light transport.

In this paper, we use covariance tracing for single scattering computations in the point-based global illumination method. We compute covariance matrices for each volume sample, and use them to predict the appropriate kernel size for single scattering computations during rendering.

3 Background

3.1 Covariance Analysis of Light Transport

Durand et al. [4] introduced a framework for frequency analysis of light transport. They compute the frequency content of the local light field around a given ray. The local light field is defined as a 4D function, with two dimensions in space and two dimensions in angle (see Figure 1). Standard operations on light transport, such as transport in free space or reflection, translate into operations on the Fourier spectrum of the local light field. Running computations with the full Fourier spectrum of the local light field is impractical. Belcour et al. [5] introduced an approximate representation for the Fourier spectrum of the local light field: the covariance matrix.

In this paper, we use the framework and notations introduced by Belcour et al. [6]. The covariance matrix is denoted as Σ. For a function defined over a 4D domain, it is a 4×4 matrix defined by:

$$\Sigma_{i,j} = \frac{1}{\int_\Omega f} \int_{x \in \Omega} (x \cdot e_i)(x \cdot e_j) f(x) dx, \forall (i,j) \in \{1, \cdots, 4\}^2, \quad (1)$$

where e_i is the i^{th} vector of the canonical basis of the 4D space Ω and $x \cdot y$ is the dot product of vectors x and y.

The eigenvectors of the covariance matrix indicate in which direction function f spreads the most and where it spreads the least; its eigenvalues are the variance of the function in all 4 principal directions. The key idea of Belcour et al. [6] is to compute the covariance matrix of the Fourier spectrum of the local light field using matrix operations corresponding to basic operations of light transport (transport in free space, reflection, occlusion). In a second step, they use these covariance matrix to adapt sampling and reconstruction to the frequency content.

Table 1 lists basic operations on the local light field: transport in free space, reflection, refraction, rotation and scaling. For each of them, we provide the corresponding operation on the covariance matrix of the Fourier spectrum. We give a detailed description below.

Transport in Free-Space When light travels in free space, the local light field undergoes a shear in space. The effect of the Fourier spectrum is a shear in angle. The covariance matrix Σ' after transport over a distance d is:

$$\Sigma' = T_d^T \Sigma T_d, \quad (2)$$

where T_d is the travel matrix for a distance of d and is defined in Table 1.

Projection When a light ray intersects with a object, we first transform the local light field into the local frame of the object: rotate the frame of the light-field to align its X axis along the intersection between the tangent plane of the light-field and the tangent plane of the object at the point of intersection, project the Y axis onto the local tangent plane and rotate to align the light-field X and Y axis to the local X and Y axis (see Figure 2). The matrix is shown in Table 1

Boundary Refraction After the light reaches the surface of participating media, we perform a series of operators: a spatial shear due to curvature; a mirror reflection; a convolution due to the

TABLE 1
Operations on the light field and their equivalence on the covariance matrix of the light field's spectra at various stages of light transport.

Transport in free space	Reflection	Refraction	Rotation by angle φ	Scaling $f(\lambda_1 x_1, \cdots, \lambda_4 x_4)$
$\Sigma' = T_d^T \Sigma T_d$	$\Sigma' = (\Sigma^{-1} + B)^{-1}$	$\Sigma' = S^T(\Sigma + w)S$	$\Sigma' = R_\varphi^T \Sigma R_\varphi$	$\Sigma' = \Lambda \Sigma \Lambda$
$T_d = \begin{bmatrix} 1 & 0 & -d & 0 \\ 0 & 1 & 0 & -d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{bmatrix} B_{\theta\theta}^{-1} \end{bmatrix}$}} \\ 0 & 0 & & \end{bmatrix}$	$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{n_1 cos(i_1)}{n_2 cos(i_2)} \end{bmatrix}$	$R_\varphi = \begin{bmatrix} cos(\varphi) & -sin(\varphi) & 0 & 0 \\ sin(\varphi) & cos(\varphi) & 0 & 0 \\ 0 & 0 & cos(\varphi) & -sin(\varphi) \\ 0 & 0 & sin(\varphi) & cos(\varphi) \end{bmatrix}$	$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix}$

irradiance cosine term. The composition of the transformations are described in Belcour et al. [5]. After these operations, we add the effect of refraction.

Refraction is handled by a convolution with the window kernel that avoids the cases when the Snell-Descartes law is no longer applicable, followed by a scale of the angular domain. The covariance matrix Σ' after refraction is:

$$\Sigma' = S^T(\Sigma + w)S. \tag{3}$$

Where w is the covariance matrix of angular window and S is the scale matrix and is defined as:

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{n_1 cos(i_1)}{n_2 cos(i_2)} \end{bmatrix}. \tag{4}$$

Where n_1, n_2 are the index of refraction of the materials on both sides of the refractive interface and i_1, i_2 are angular parameters.

After refraction, we perform another set of operators to switch the light filed from local frame to world frame.

Medium travel The refracted light ray travels in the medium and stops at each sampled position, called volume sample. The covariance matrix is also updated by the travel d_i distance, where d_i is the distance traveled from the entry point at the surface to the volume sample v_i.

The light ray might get reflected inside the medium, probably, many times. The covariance matrix is updated correspondingly, performing projection, curvature and symmetry operators before reflection, similar to the refraction. The covariance matrix Σ' after reflection is:

$$\Sigma' = \left(\Sigma^{-1} + B\right)^{-1}, \tag{5}$$

where B is is the pseudo-inverse of the covariance matrix and is defined in Table 1.

3.2 Point Based Light Transport In Participating Media

Wang et al. [3] introduced a point based method for light transport in participating media with refractive boundaries. As all point-based global illumination methods, it works in two steps: in a *precomputation step*, they distribute surface samples on the boundary of the media, refract the light rays on these surface samples into the volume and sample the refracted light ray in the volume to get volume samples. Volume samples and surface samples are organized into two separate hierarchies for further queries (see Figure 3). For the *rendering step*, they separate scattering effects into *single*, *double* and *multiple* scattering, depending on the number of scattering events inside the medium. Each of these effects are computed separately. During the rendering

TABLE 2
Notations.

Material properties	
σ_a	absorption coefficient
σ_s	scattering coefficient
$\sigma_t = \sigma_s + \sigma_a$	extinction coefficient
$\ell = 1/\sigma_t$	mean free path
$\alpha = \sigma_s/\sigma_t$	single scattering albedo
p	phase function
g	mean cosine of phase function
η	media refractive index
Volume samples	
$\vec{x}_v$	position
$\vec{d}_v$	direction of incoming light
I_v	intensity of node v
f_v	frequency factor of node v
S_v	surface area of node v
A_v	area of the octree leaf cell
Camera ray samples	
P_k	sample point on camera ray
d_k	depth of sample point P_k

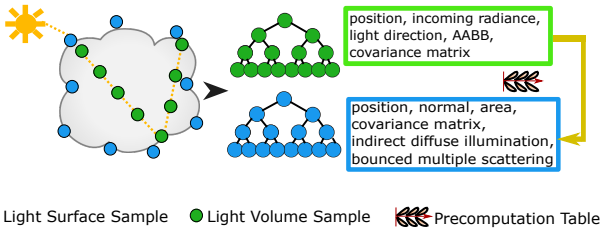


Fig. 3. Precomputation Step: first, we distribute surface samples at the volume surface. Then we trace the refracted light from the surface sample into the volume and sample the rays inside the volume to get volume samples. Surface samples and volume samples are organized in two separate hierarchies for further queries in the rendering step.

step, camera rays are refracted into the volume and are sampled into camera samples. They then compute the contributions from volume samples to camera samples. Double and multiple scattering use a tree cut of the hierarchy of volume samples, and are already efficient. For single scattering, they traverse the hierarchy of volume samples until they have all leaf nodes that are close to the camera sample P_k (see Figure 4).

$$L(P_k) = e^{-\sigma_t d_k} \frac{1}{N} \sum_{v_i} \sigma_s p(\mathbf{o}, \mathbf{d_{v_i}}) I_{v_i}. \tag{6}$$

The details for other types of effects can be found in Wang

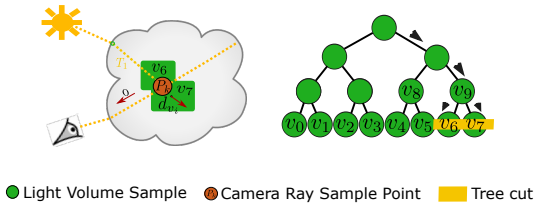


Fig. 4. Rendering Step (Single scattering only): we traverse the volume sample tree from the top. If the AABBs of the nodes contain the current camera sample, the child nodes are further traversed for AABB test until reaching leaf nodes, otherwise, the nodes are discarded. For all the volume samples included in each found leaf node, if its orientated bounding box contains the camera sample, this volume sample contributes to the camera sample.

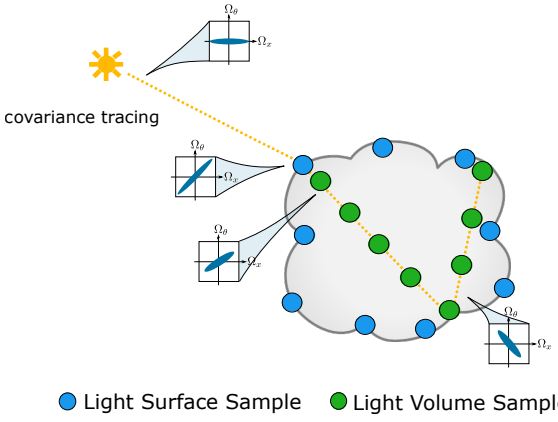


Fig. 5. Covariance tracing from light source to volume samples. The covariance matrix is created from the light source, updated for each operator along the path from the light to volume samples and stored in each volume sample.

et al. [3]. Single scattering is a complicated effect: it can be very high frequency in localized areas, called *volume caustics*, and low frequency in other areas. Predicting the position of the volume caustics is impossible; as a consequence, Wang et al. [3] had to use a large number of volume samples, and always used the leaves of the hierarchy to compute single scattering. It produced the expected high-frequency volume caustics but with a large memory footprint and long computation time for the precomputation step. In this paper, we use Frequency Analysis tools to predict the location of volume caustics, and adapt sampling accordingly.

4 Covariance Driven Single Scattering Computation

4.1 Context

Our algorithm is an extension of Wang et al. [3] algorithm for Point Based Light Transport in participating media with refractive boundaries. We keep the original algorithm for double and multiple scattering, and only change the method for single scattering. Regarding *single*, *double* and *multiple* scattering, we only count the number of scattering events, independently of the number of internal reflections on the specular surface. As in the original paper, we focus on homogeneous participating media with refractive boundaries.

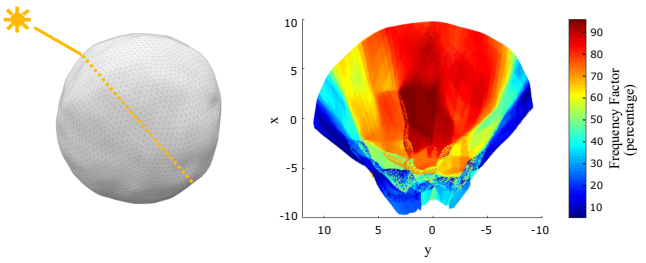


Fig. 6. Visualization of the frequency factor of the volume samples for the Bumpy Sphere Scene. The frequency factor is the determinant of the covariance matrix, and it is connected to the frequency content of the local light field. The larger the value is, the higher the frequency is. We store the frequency factor for the volume sample.

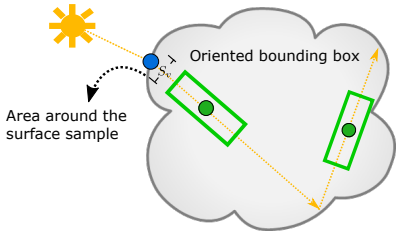


Fig. 7. The kernel shape of a volume sample: oriented bounding box.

4.2 Covariance Tracing for Volume Samples

During the *precomputation step*, we distribute volume samples along the refracted light ray from the surface sample (see Figure 5). We compute the covariance matrix along with the radiance information for each ray. We create an initial covariance matrix Σ at the light source and update the covariance matrix with the following light transport operators: *Free space travel*, *Projection*, *Boundary Refraction* and *Medium travel*. At each volume sample, we compute and store the covariance matrix's determinant, denoted as η, and defined by:

$$\eta = \sqrt{|\Sigma|}. \tag{7}$$

η goes from 0 to 1. The higher the value of η, the larger the frequency content at this location. $\eta = 0$ corresponds to a uniform, constant distribution (low frequency), $\eta = 1$ corresponds to a Dirac (high frequency). These volume samples, with the determinant of their covariance matrix, are organized into a spatial hierarchy, as in Wang et al. [3]. Figure 6 shows the spatial distribution of the determinant of covariance matrix for the volume samples inside the bumpy sphere scene.

4.3 Kernel Prediction with Covariance

During rendering step, the camera ray is refracted into the volume and is sampled regularly into camera samples. For each camera sample, the volume spatial hierarchy is traversed from the top, checking whether the axis aligned bounding box (AABB) of the nodes contains the queried camera sample. If the AABB of a node does not contain the camera sample, the node gets discarded, otherwise, we keep descending recursively in the hierarchy until reaching the leaf nodes. For each leaf node whose AABB contains the camera sample, we perform a more accurate kernel test using the oriented bounding box (OBB) (see Figure 7). The oriented

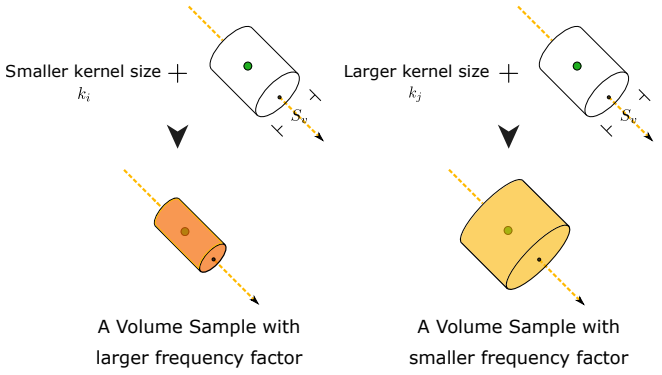


Fig. 8. The kernel size of a volume sample is adjusted using the frequency factor from the covariance prediction. Besides the kernel size, the radiance of the volume sample is updated correspondingly: the left volume sample has higher radiance.

bounding box is a cylinder whose axis is the light ray inside the material. The cylinder length is constant, related to the sampling on the light ray. We adapt the cylinder cross section depending on the frequency content η for this volume sample: smaller cross section for high frequency content, larger cross section for low frequency content. Specifically, we multiply the cross-section s_v of the Oriented Bounding Box with a multiplier k_i (see Figure 8):

$$k_i = C_k(\alpha cos(\eta\pi) + (1-\alpha)), 0 < \alpha < 1, \tag{8}$$

where C_k is a constant factor to control the kernel size, set as $20 \times \text{clamp}(\sigma_t, 0.3, 0.8)$ and α is a tweakable parameter, set as 0.4 in practice. For high frequencies (large values of η), we get $k_i \approx C_k(1-\alpha)$, resulting in a lower kernel size; for low frequencies, we get $k_i \approx C_k$, resulting in a larger kernel size. This allows to preserve the sharp details in volume caustics.

The radiance of the volume sample is also updated correspondingly, multiplying with $\frac{1}{k_i}$, to keep conservation of energy. In the end, we use the Equation 6 to evaluate single scattering radiance. With this kernel prediction aware of covariance of local light field, we achieve the sharp effects associated with single scattering using fewer samples (see Table 4).

5 Result

We have implemented our algorithm inside the Mitsuba Renderer [14]. The covariance tracing is based on the implementation by Belcour [15]. We compared our algorithm against (i) Wang et al. [3], (ii) Holzschuch [2] and (iii) Unified points, beams and paths (UPBP) by Křivánek et al. [16] which we consider as the reference for quality validation. We use the SmallUPBP implementation by Křivánek et al. [17] for UPBP.

All timings in this section are measured on a 2.20GHz Intel Xeon (16 cores) with 24 GB of main memory. All pictures were rendered at a resolution of 512×512 pixels. We measure numerical differences with the Mean-Squared Error (MSE). All materials in our test scenes are homogenous materials, with Henyey-Greenstein phase functions and smooth refractive boundaries. Material properties are from Křivánek et al. [16] and Holzschuch [2] (see Table 3).

TABLE 3
Material parameters.

Name	α			ℓ			g
	R	G	B	R	G	B	
BmpS.	0.955	0.677	0.457	4.546	3.226	2.174	0.9
Bunny	0.165	0.115	0.073	8.231	4.751	2.269	0.9
Buddha	0.240	0.480	0.960	2.000	2.000	2.000	0.8
Oil	0.004	0.454	0.100	9.709	11.628	2.740	0.9

5.1 Quality Validation

We first compare our method with Wang et al. [3], Holzschuch [2] and reference images computed using UPBP [16] in Figures 9, 10 and 11 with single scattering only. Qualitatively, our approach produces results that are visually identical to the ground truth (rendered with UPBP). The original method by Wang et al. [3] produces over-blurred results with equal time and costs twice as much with equal quality. The method by Holzschuch [2] is highly dependent on the scene complexity, so it is extremely expensive on the Bunny and Buddha Scene.

We also provide difference (x5) images in Figures 9, 10 and 11. These difference images also demonstrate that our algorithm provide higher quality than Wang et al. [3] with equal computation time. The exact error represented with MSE can be found in Table 4.

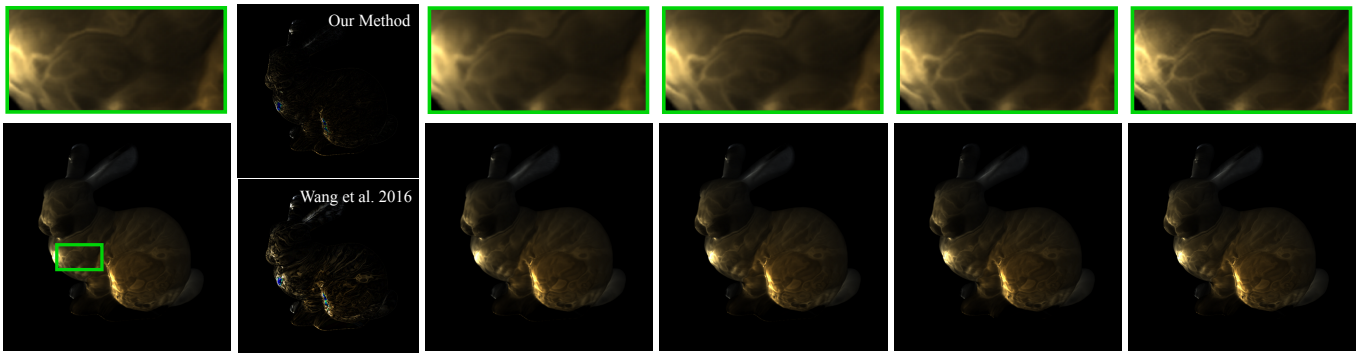
Figure 12 corresponds to a complex scene, with several layers outside the participating medium. We rendered these images with a full solution, including single scattering, double scattering and multiple scattering. We compare with the method by Wang et al. [3] and UPBP. Compared to UPBP, our method provide almost visually identical result. With equal time, Wang et al. [3] produces result that loses details. Our method costs only half the time compared to Wang et al. with equal quality. One key advantage of our method is that we can reduce the number of samples required to get accurate single scattering computations. In this experiment, we find that this reduced number of samples has no impact on the quality of double and multiple scattering computations.

5.2 Performance Measurement

Table 4 displays the parameters, settings and timings for all our test scenes. We set the maximum path depth for the first three scenes as 4, to compute single scattering only. The larger pixel sample count for Buddha scene is for anti-aliasing. The Oil Scene requires more pixel sample count to converge. The table also reports the MSE between reference pictures and pictures generated with our rendering algorithm. This quantitative data confirms the qualitative visual impression: the pictures generated by our algorithm are very close to the reference images, and better than the pictures generated by the unmodified algorithms with equal time. Our method only uses one fifth of the memory cost of Wang et al. [3] for equal quality requirements.

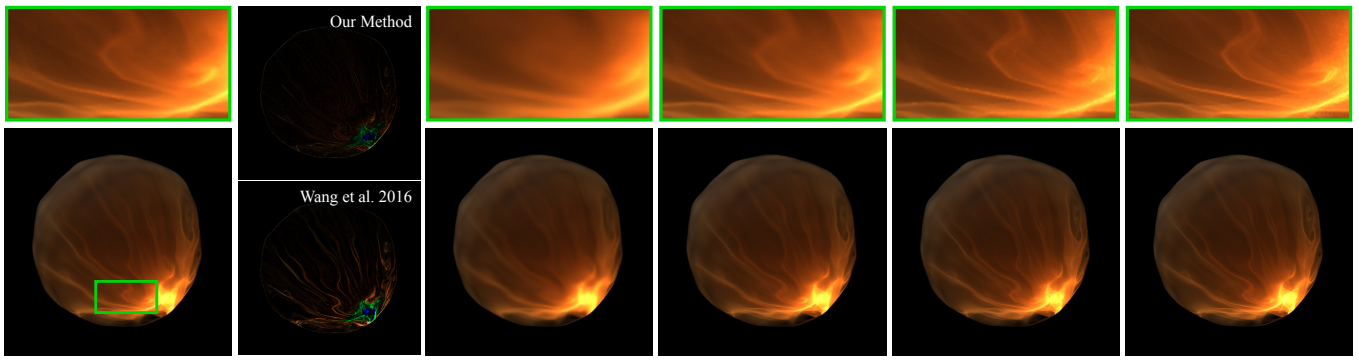
Our method is faster than the reference Point Based light transport method by Wang et al. [3] for equal quality requirements, approximately twice as fast.

By analyzing both the equal time and equal quality comparison with Wang et al. [3], we observe our method is faster than Wang et al. [3] with same sample count. The same sample count results in identical pre-processing time. But our method has less rendering



(a) Our Method, 37.71s | difference (x5) | (b) Wang et al. 2016, Equal time, 37.90s | (c) Wang et al. 2016, Equal quality, 1.15m | (d) Holzschuch 2015, 33.75m | (e) UPBP, Reference, 6h

Fig. 9. Our algorithm (a), compared to: the original Point based light transport method by Wang et al. [3] ((b) and (c)), Accurate Single Scattering by Holzschuch [2] (d) and a reference image using UPBP (e) on the Stanford bunny. Using covariance tracing for kernel size predicting (a) produces almost identical results to the ground truth (e), and is much faster than the original method (c) with equal quality and provides higher quality (b) with equal time. Material: Bunny. The difference images are obtained by comparing our method (a) and Wang et al. [3] using equal computation time (b) with the reference image from UPBP (e). Single Scattering Only.



(a) Our Method, 26.47s | difference (x5) | (b) Wang et al. 2016, Equal time, 26.81s | (c) Wang et al. 2016, Equal quality, 1.35m | (d) Holzschuch 2015, 2.98m | (e) UPBP, Reference, 18h

Fig. 10. Our algorithm (a), compared to: the original Point based light transport method by Wang et al. [3] ((b) and (c)), Accurate Single Scattering by Holzschuch [2] (d) and a reference image using UPBP (e) on the bumpy sphere scene. Using covariance tracing for kernel size predicting (a) produces almost identical results to the ground truth (e), and is much faster than the original method (c) with equal quality and provides higher quality (b) with equal time. Material: Bumpy Sphere. The difference images are obtained by comparing our method (a) and Wang et al. [3] using equal computation time (b) with the reference image from UPBP (e). Single Scattering Only.

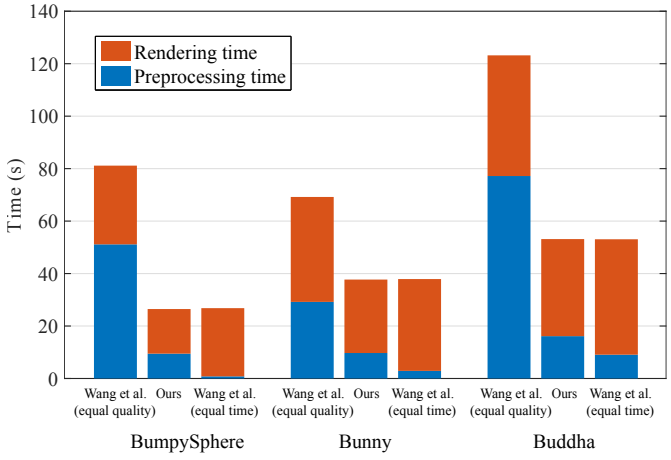


Fig. 13. Preprocessing time and rendering time of Wang et al. [3] and our method as a Stacked-Column on the Bumpysphere, Bunny and Buddha Scene.

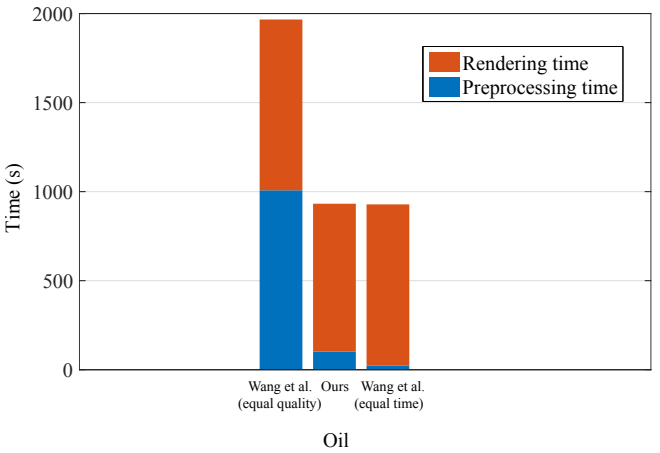
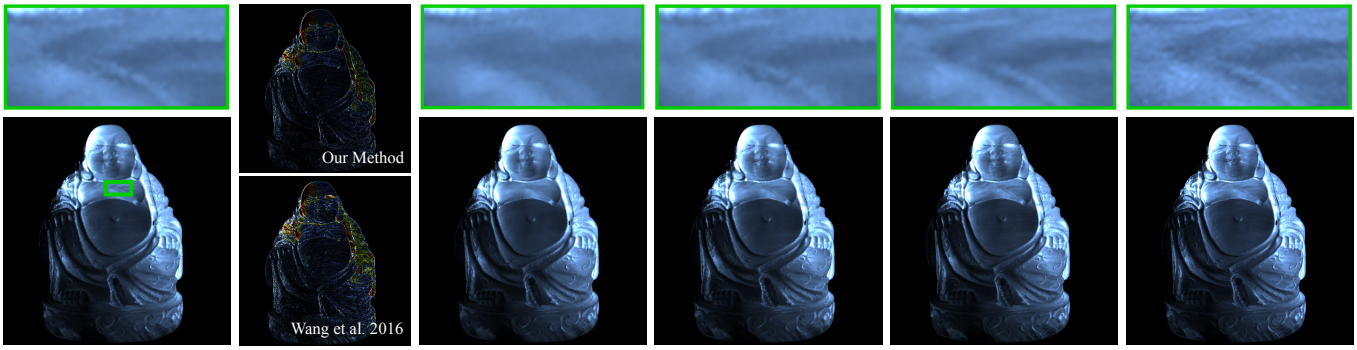
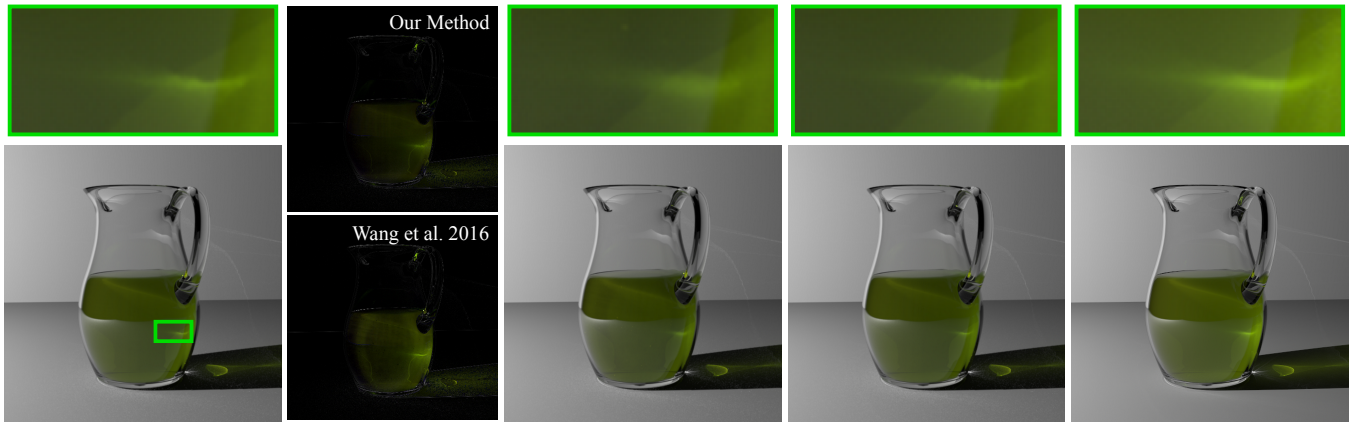


Fig. 14. Preprocessing time and rendering time of Wang et al. [3] and our method as a Stacked-Column on the Oil Scene.



(a) Our Method, 53.14s | difference (x5) | (b) Wang et al. 2016, Equal time, 53.07s | (c) Wang et al. 2016, Equal quality, 2.05m | (d) Holzschuch 2015, 4.13h | (e) UPBP, Reference, 9h

Fig. 11. Our algorithm (a), compared to: the original Point based light transport method by Wang et al. [3] ((b) and (c)), Accurate Single Scattering by Holzschuch [2] (d) and a reference image using UPBP (e) on the buddha scene. Using covariance tracing for kernel size predicting (a) produces almost identical results to the ground truth (e), and is much faster than the original method (c) with equal quality and provides higher quality (b) with equal time. Material: buddha. The difference images are obtained by comparing our method (a) and Wang et al. [3] using equal computation time (b) with the reference image from UPBP (e). Single Scattering Only.



(a) Our Method, 15.53m | difference (x3) | (b) Wang et al. 2016, Equal time, 15.47m | (c) Wang et al. 2016, Equal quality, 32.78m | (d) UPBP, Reference, 6h

Fig. 12. Our algorithm (a), compared to: the original Point based light transport method by Wang et al. [3] ((b) and (c)), Accurate Single Scattering by Holzschuch [2] (d) and a reference image using UPBP (e) on the oil scene. Using covariance tracing for kernel size predicting (a) produces almost identical results to the ground truth (e), and is much faster than the original method (c) with equal quality and provides higher quality (b) with equal time. Material: Oil. The difference images are obtained by comparing our method (a) and Wang et al. [3] using equal computation time (b) with the reference image from UPBP (e). Full solution including single scattering, double scattering and multiple scattering.

cost, as our kernel size is more compact, thanks to the covariance driven model. A smaller kernel size results in tighter bounding box around samples and the tree nodes, so the uncontributed nodes can be discarded sooner.

Figures 13 and 14 show the computation time for our test scenes, with the breakdown between pre-processing (computing the surface and volume samples) and rendering. Regarding the equal quality comparison, most of the gains come from a reduced pre-processing computation time; pre-processing time is divided by 3 for the bunny scene, by 10 for the oil scene. There are also substantial gains in rendering time, 20 to 30 % , due to the compact kernel size. The overall reduction in computation time makes our method approximately twice as fast as the original method. Regarding the equal time comparison, our method has longer pre-processing cost due to more surface and volume samples and shorter rendering cost, as our method has more compact kernel size.

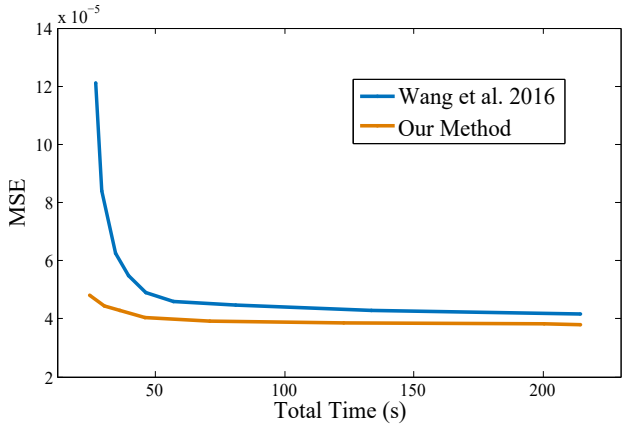


Fig. 15. Mean Square Error as a function of rendering time for our method and Wang et al. [3] on the bumpy sphere Scene.

TABLE 4
Computation time and memory costs for our test scenes. vs#: number of volume samples. Pre. time: preprocessing time (computing volume and surface samples). Rend. time: rendering time. Total computation time is a sum of these two.

Scene	Pixel Sample Count	UPBP	Holzschuch 2015	Wang et al. 2016												Our Method					
				Equal time						Equal quality											
		Rend. (h)	Rend. (m)	vs# (K)	Mem. (MB)	Pre. (s)	Rend. (s)	Total. (s)	Error MSE	vs# (M)	Mem. (GB)	Pre. (s)	Rend. (s)	Total. (s)	Error MSE	vs# (M)	Mem. (MB)	Pre. (s)	Rend. (s)	Total. (s)	Error MSE
BumpyS.	4	18	2.98	368.42	45.0	0.81	26.00	26.81	1.21e-4	15.51	1.8	51.14	30.00	81.14	4.46e-5	4.19	507.6	9.47	17.00	26.47	4.47e-5
Bunny	16	6	33.75	988.47	120.7	2.90	35.00	37.90	5.505e-5	12.08	1.4	29.20	40.00	69.20	3.24e-5	3.94	481.4	9.71	28.00	37.71	3.26e-5
Buddha	256	9	248.05	3163.59	386.2	9.07	44.00	53.07	7.83e-4	21.99	2.6	77.18	46.00	123.18	6.49e-4	7.34	896.5	16.14	37.00	53.14	6.50e-4
Oil	8192	6	-	1345.03	164.2	22.43	906.00	928.43	3.62e-4	71.88	8.6	1007.53	959.00	1966.53	2.84e-4	10.88	1331.2	102.15	830.00	932.15	2.85e-4

5.3 Parameter Analysis

Figure 15 shows quality as a function of rendering time for our method and Wang et al. [3], for the bumpy sphere Scene. We gradually increase the number of volume samples for both methods, resulting in better quality at the expense of computation time. We compare both methods with the reference solution computed using UPBP, and measure the Mean Square Error between each picture and the reference. Our method provides consistently better quality than the original method. The relative speedup is especially impressive for a low number of volume samples (corresponding to a rendering time of less than 50 s). As we increase the number of volume samples, both methods reach a plateau in terms of quality, with our method providing higher quality results.

6 Conclusion

We have presented a method that combines frequency analysis of light transport using covariance matrix with point-based light transport algorithm for single scattering effects in participating media with refractive boundaries. Combining frequency analysis of light transport with point-based illumination methods is done naturally by computing the covariance matrix for each volume sample, and storing its determinant. We use this precomputed determinant to predict the kernel size for each volume sample, resulting in a significant speedup for the same quality. Most of the speedup comes from the precomputation stage: using frequency analysis lets us reduce the number of volume samples without loss of quality. A second speedup comes from the rendering time, where a smaller number of samples makes hierarchy traversal faster.

This first approach shows the benefits of combining frequency analysis of light transport with point-based global illumination: computing frequency information can be done during the illumination samples computation. Re-using this information during the rendering phase can provide significant speedups. The benefits are important for high-frequency effects, which can require many illumination samples for convergence. In future work, we want to investigate a more complete use of frequency analysis tools for point-based global illumination (PBGI).

Acknowledgments

We thank the reviewers for the valuable comments. This work has been partially supported by the National Key R&D Program of China under grant No. 2017YFB0203000, the National Natural Science Foundation of China under grant No. 61802187, 61872223, the Natural Science Foundation of Jiangsu under grant No. BK20170857, the fundamental research funds for the central universities No. 30918011320, ANR project ANR-15-CE38-0005 "Materials" and the Young Scholars Program of Shandong University under grant No.2015WLJH41.

References

[1] B. Walter, S. Zhao, N. Holzschuch, and K. Bala, "Single scattering in refractive media with triangle mesh boundaries," *ACM Transactions on Graphics*, vol. 28, no. 3, aug 2009. 1
[2] N. Holzschuch, "Accurate computation of single scattering in participating media with refractive boundaries," *Computer Graphics Forum*, vol. 34, no. 6, pp. 48–59, Sep. 2015. 1, 5, 6, 7
[3] B. Wang, J.-D. Gascuel, and N. Holzschuch, "Point-Based Light Transport for Participating Media with Refractive Boundaries," in *Eurographics Symposium on Rendering 2016 - EI&I*, Dublin, Ireland, Jun. 2016. 1, 2, 3, 4, 5, 6, 7, 8
[4] F. Durand, N. Holzschuch, C. Soler, E. Chan, and F. Sillion, "A frequency analysis of light transport," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005)*, vol. 24, no. 3, aug 2005. 1, 2
[5] L. Belcour, C. Soler, K. Subr, N. Holzschuch, and F. Durand, "5D covariance tracing for efficient defocus and motion blur," *ACM Trans. Graph.*, vol. 32, no. 3, Jun. 2013. 1, 2, 3
[6] L. Belcour, K. Bala, and C. Soler, "A local frequency analysis of light scattering and absorption," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 163:1–163:17, Sep. 2014. 1, 2
[7] P. H. Christensen, "Point-based approximate color bleeding," *Technical Report*, vol. 2, 2008. 1
[8] A. Arbree, B. Walter, and K. Bala, "Single-pass scalable subsurface rendering with lightcuts," in *Computer Graphics Forum*, 2008, pp. 507–516. 2
[9] H. Igehy, "Tracing ray differentials," in *Annual Conference on Computer Graphics*, 1999, pp. 179–186. 2
[10] F. Suykens and Y. D. Willems, *Path Differentials and Applications*. Springer Vienna, 2001. 2
[11] C. Soler, K. Subr, and N. Holzschuch, "Fourier depth of field," *Acm Transactions on Graphics*, vol. 28, no. 2, pp. 1–12, 2009. 2
[12] K. Egan, Y. T. Tseng, N. Holzschuch, and R. Ramamoorthi, "Frequency analysis and sheared reconstruction for rendering motion blur," *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, pp. 1–13, 2009. 2
[13] L. Belcour, L.-Q. Yan, R. Ramamoorthi, and D. Nowrouzezahrai, "Antialiasing Complex Global Illumination Effects in Path-space," *ACM Transactions on Graphics*, vol. 36, no. 1, 2017. 2
[14] W. Jakob, "Mitsuba renderer," http://www.mitsuba-renderer.org/, 2010. 5
[15] L. Belcour, "Covariance tracing source code." https://github.com/belcour/CovarianceTracing., 2016. 5
[16] J. Křivánek, I. Georgiev, T. Hachisuka, P. Vévoda, M. Šik, D. Nowrouzezahrai, and W. Jarosz, "Unifying points, beams, and paths in volumetric light transport simulation," *ACM Trans. Graph. (proc. SIGGRAPH)*, vol. 33, no. 4, pp. 1–13, Aug. 2014. 5
[17] J. Křivánek, "SmallUPBP," http://www.smallupbp.com/, 2014. 5

Yulin Liang is a postgraduate at School of Software, Shandong University. He received his Bachelor from Shandong University in 2016. He studied for a master's degree from 2016 and is working on rendering.

Beibei Wang is an Associate Professor at Nanjing University of Science and Technology. She received her PhD from Shandong University in 2014 and visited Telecom ParisTech from 2012 to 2014. She worked as a Postdoc in Inria from 2015 to 2017. She joined NJUST in March 2017. Her research interests include rendering and game development.

Lu Wang is a Professor at School of Software, Shandong University. She received her PhD from Shandong University in 2009. Her research interests include photorealistic rendering and high performance rendering.

Nicolas Holzschuch is a Senior Researcher at INRIA Grenoble Rhône-Alpes, and the scientific leader of the MAVERICK research team. He received his PhD from Grenoble University in 1996 and his Habilitation in 2007. He joined INRIA in 1997. His research interests include photorealistic rendering and real-time rendering, with an emphasis on material models and participating media.