

Bidirectional Plateau-Border Scattering Distribution Function for Realistic and Efficient Foam Rendering

R. Li^{1,2} , X. Liu³ , R. Wang¹ , P. Shen⁴ , L. Liu^{1,2} , and B. Wang⁵ 

¹University of Science and Technology of China, China

²Laoshan Laboratory, Qingdao, Shandong, China

³Beihang University, China

⁴The University of Hong Kong, China

⁵Nanjing University, China

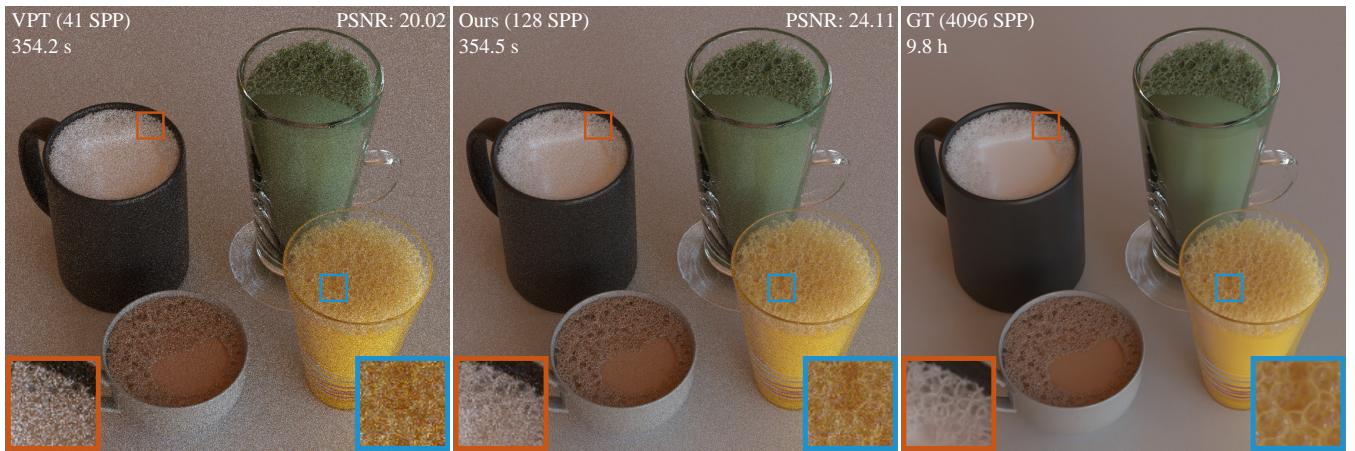


Figure 1: We introduce the Plateau border structure into liquid foam rendering at the geometric level. We then propose a bidirectional Plateau-border scattering distribution function (BPSDF) to aggregate the scattering within a Plateau border and shorten the path length. By designing a neural representation of the BPSDF, our model can render liquid foams both realistically and efficiently. Our BPSDF can significantly reduce noise compared to volume rendering across various media, from high-scattering to high-absorption and from isotropic to anisotropic media.

Abstract

Liquid foams are a common phenomenon in our daily life. In computer graphics, rendering realistic foams remains challenging due to their complex geometry and light interactions within the foam. While the structure of the liquid foams has been well studied in the field of physics, it's rarely leveraged for rendering, even though it is essential for achieving realistic appearances. In physics, the intersection of two bubbles creates a liquid-carrying channel known as the Plateau border (PB). In this paper, we introduce the Plateau border into liquid foam rendering by explicitly modeling it at the geometric level. Although modeling of PBs enhances visual realism with path tracing, it suffers from extensive rendering costs due to multiple scattering effects within the medium contained in the PB. To tackle this, we propose a novel scattering function that models the aggregation of scattering within the medium surrounded by a Plateau border, termed the bidirectional Plateau-border scattering distribution function (BPSDF). Since no analytical formulation can be derived for the BPSDF, we propose a neural representation, together with importance sampling and probability distribution functions, to enable Monte Carlo-based rendering. By integrating our BPSDF into path tracing, our method achieves both realistic and efficient rendering of liquid foams, producing images with high fidelity.

CCS Concepts

- Computing methodologies → *Rendering; Reflectance modeling;*

1. Introduction

Liquid foams are common natural phenomena in daily life, such as the bubbles on top of beverages. Simulating and rendering these phenomena are both important and fascinating in the fields of physics and computer graphics. Unfortunately, realistically rendering liquid foams remains challenging due to their unique structure and the complex interactions of light as it travels through the bubble films and surrounding medium. In this paper, we address the issue of realistically and efficiently rendering a collection of bubbles in a liquid medium.

Existing bubble rendering approaches [IMN04; GSP*19; Mor03] have made efforts on thin-film rendering of individual bubbles, taking into account light interference, while another set of works [BSW10; vdLGS09] focus on real-time rendering of fluids in screen space. However, neither can render liquid foam realistically, as its visual appearance is highly related to the foam structure. At the intersection of two bubbles, three films converge in a line to form a liquid-carrying channel known as a Plateau border (PB) (see Fig. 2) in physics. The PBs are crucial for both the dynamic behavior and visual characteristics of foams, as they contain most of the liquid in the foam. The importance of PB for liquid foam rendering has been identified by Kück et al. [KVG02] and Greenwood et al. [GH04]. Unfortunately, these works approximate the visual effects of Plateau borders using a simplified shading model, rather than accurately modeling the detailed geometry, resulting in less realistic renderings.

In this paper, we target high-fidelity liquid foam rendering while maintaining acceptable time cost. For this, we propose to model the Plateau border structure explicitly between bubbles, where the cross-sectional shape is represented by three circular arcs [LL65; KR87; BKD*08] (see Fig. 2) following Plateau’s laws in the field of physics. To our knowledge, this is the first time the Plateau border structure has been modeled explicitly for liquid foam rendering. While this explicit geometric modeling of PBs can bring high level of realism with Monte Carlo rendering, it suffers from extensive rendering costs due to the medium gathered in the Plateau border. To this end, we propose a novel scattering function for a Plateau border, referred to as the *bidirectional Plateau-border scattering distribution function (BPSDF)*. This function models the scattering that occurs within the medium encircled by the Plateau border. Since there is no existing analytical formulation, we propose a neural representation of the BPSDF, which is parameterized on the PB shape, medium properties, and ray queries (including both incoming and outgoing positions and directions). Specifically, we design a set of lightweight neural functions for BPSDF evaluation, importance sampling, and probability distribution function (PDF) evaluation. Thanks to this neural representation of the Plateau border, we can efficiently aggregate scattering within a Plateau border, significantly reducing path lengths during Monte Carlo rendering. Consequentially, the liquid foam can be rendered both realistically and efficiently. Note that our work mainly focuses on the rendering of liquid foams, and the simulation of foam behaviors is beyond our scope. We position our work in the geometric optics domain, although the interference caused by thin films is orthogonal to our work and can be integrated directly. To summarize, our main contributions include:

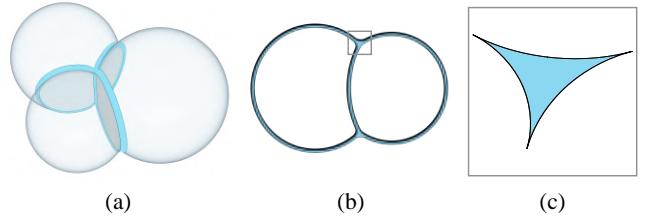


Figure 2: Illustration of intersecting bubbles and PBs in (a) 3D and (b) 2D, with a zoom-in for the cross-section of a PB in (c).

- a realistic liquid foam rendering approach by explicitly modeling the Plateau border structure,
- a novel scattering function, BPSDF, defined for Plateau borders that aggregates scattering and simplifies light transport in liquid foam, and
- a neural representation of the BPSDF that facilitates efficient evaluation, importance sampling, and PDF computation, enabling effective light transport in liquid foam.

2. Related work

Foam physics. Exploring the physics of foam has attracted mathematicians and physicists for centuries. Foams in daily life exhibit significantly different geometric and visual behaviors due to the complex physics and properties of different liquids. In general, foams can be classified into dry foams and wet foams by the liquid fraction, i.e. the amount of liquid contained in the foam [WH00; CCE*13]. For dry foams that contain little liquid, the Belgian physicist Joseph Plateau developed the famous Plateau equilibrium rules to describe their structures [Pla73]. Wet foams are more difficult to model since there is sufficient liquid to deform the shapes of bubbles and form more complex structures.

The faces of bubbles are thin films that are curved because of pressure differences. According to Plateau’s rules, the films intersect in threes and form liquid-carrying channels, which are known as Plateau borders (Fig. 2). Previous works approximated the cross-section of a Plateau border as a triangle with concave sides, and each side is a circular arc [CCE*13; Ngu02; GZC04]. The real shape of a Plateau border is hard to determine because of the surface tension and gravity of the liquid [STW08; Bra92; XWPA17]. We make the same approximation about the Plateau border’s shape as previous works and leave the modeling of more complex shapes as future works.

Foam simulation and rendering. The simulation and rendering of bubbles and foams in computer graphics have attracted considerable attention. Simulation efforts range from capturing thin film thickness [HIK*20; DWK*22; ISN*20] to animating foam flow dynamics [IYAH17; WFS22; QLY*23]. In the rendering field, main efforts have been made on light interference from soap film thickness [Mor03; IMN04; GSP*19] and real-time foam rendering in the screen space [vdLGS09; BSW10; DCGG11]. Droske et al. [DHV*23] introduce path tracing in water and some special appearances like bubbles.

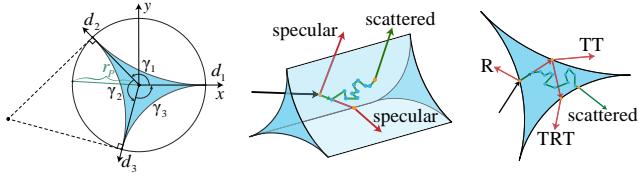


Figure 3: Left: the cross-section of a PB is modeled as three tangent circular arcs. Middle: within a PB, there are two types of light paths: the specular path (red), which consists of surface interactions only, and the scattered path (green), which includes at least one scattering event within the medium. Right: different types of light paths within a PB. Surface events are denoted by orange dots, and medium events are denoted by blue dots.

The closest work to ours is by Kück et al. [KVG02] and Greenwood et al. [GH04]. They identified the impact of Plateau borders on the characteristic visual properties of liquid foams. However, they modeled the scattering in Plateau borders by utilizing an empirical shading model, which sacrificed visual realism and richness of liquid types. In contrast, our method models Plateau borders explicitly at the geometric level, which enhances realism and allows for rendering various foam types by controlling the medium in the PB.

Neural appearances. Neural networks are widely used in material representation. Many works focus on using neural networks for material representation and enhancing the effectiveness and richness [FR22; TUGM22; XZJM24; SRRW21; KMX*21; KWM*22a]. Neural representations are utilized for a wide range of materials, like measured BRDF [HGC*20; ZZW*21], layered materials [WJHY22; FWH*22; GLH*23; ZRW*24], woven fabric [CWW24] and fur [ZZW*22]. There are also many works using neural compression of the bidirectional texture function (BTF) [RGW19; RGJW20; FWH*23; KWM*22b; XMWY24]. In addition, the importance sampling technique for neural materials has attracted considerable attention in recent studies [XWH*23; FBL*24; LHL*24]. Neural networks also have broad applications in simulating subsurface events [VKJ19; TTJ*24; TFRJ24; LGH*24; ZK24; LHW21] and volume rendering [HYL*23; KMM*17]. Different from these works, our method designs neural networks specialized for the PB structure, considering its characteristics, including the evaluation, importance sampling, and PDFs.

3. Preliminaries

In this section, we briefly review the modeling of Plateau borders (Sec. 3.1) and the optical effects in liquid foam (Sec. 3.2).

3.1. Plateau border modeling

Accurately computing the shape of a PB is challenging, as it requires solving for factors such as surface tension and gravity. In physics, a common method to approximate the cross-section of a PB is to use three tangent circular arcs, as shown in Fig. 2 [LL65; KR87; BKD*08], as a PB is formed by the intersection of three films.

A PB can be determined by its cross-section and a Plateau axis, which is perpendicular to the cross-section. The center of the PB cross-section is identified as the incenter of the triangle formed by centers of the three tangent circles, and its radius is represented as r_P , which remains constant along the Plateau axis. The lines drawn from the center to the three tangent points define three directions, denoted as d_1 , d_2 , and d_3 . Each pair of these directions has an angle between them, defined as γ_1 , γ_2 , and $\gamma_3 = 2\pi - \gamma_1 - \gamma_2$ (see Fig. 3). For a finite-length PB, the center of the cross-section at the midpoint along the Plateau axis determines the PB’s center.

3.2. Optical effects in liquid foam

There are two components in a liquid foam: the bubble film and PB. A PB is made of three surfaces filled with a liquid medium. A light ray can either be reflected or refracted at the film surface. When a light ray enters the liquid, the average distance between collisions of light and particles is determined by the medium’s mean free path (mfp) l . The light can be scattered or absorbed, depending on the absorption and scattering coefficients of the medium. A phase function expresses the scattered direction of a light ray. A light ray may bounce numerous times within a PB before it eventually exits (see Fig. 3).

4. Bidirectional Plateau-border scattering distribution function

By treating Plateau borders as geometrical shapes and applying a specular shading model to their surfaces while modeling the interior as a liquid medium, we can render liquid foams using Monte Carlo-based techniques, such as path tracing. As detailed in Sec. 3.2, numerous scattering events occur within the Plateau border’s medium. To avoid extensive scattering events, we propose a scattering distribution function specifically for the Plateau border, which aggregates the scatterings within the Plateau border. This function is termed *bidirectional Plateau-border scattering distribution function (BPSDF)*, analogous to the bidirectional curve scattering function [MJC*03] defined for hairs. We first parametrize BPSDF in Sec. 4.1 and then introduce its full formulation in Sec. 4.2.

4.1. BPSDF parameterization

We start by constructing a coordinate frame for a PB, and then introduce its parameters. Following the symbols defined in Sec. 3.1, by setting $\gamma_1 \geq \gamma_2 \geq \gamma_3$, we define the PB center as the origin, d_1 as the x axis, the cross product of d_1 and d_2 as z axis and product of z axis and x axis as y axis, as illustrated in Fig. 4. Note that we assume the PB is straight; if not, we discretize it into several straight segments.

The parameters of a PB consist of the medium properties M , the Plateau border shape parameters P , and the ray query (incoming and outgoing positions and directions) parameters:

$$M = \{\eta, l, \alpha_s, g\}, \quad (1)$$

$$P = \{\gamma_1, \gamma_2, \gamma_3, r_P\}, \quad (2)$$

$$Q = \{x_i, \omega_i, x_o, \omega_o\}, \quad (3)$$

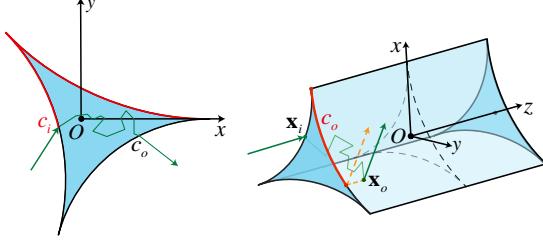


Figure 4: Left: parameterization for positions on the cross-section. The arc length parameter starts from the tangent point at the x axis and increases counterclockwise to describe the distance along the boundary of the PB's cross-section. Right: the coordinate frame defined for a PB. Our method assumes no position offset along the PB axis (orange) between the incoming point \mathbf{x}_i and outgoing point \mathbf{x}_o , because the length of PB along the PB axis is much larger than its width.

where η is the index of refraction (IOR), l and α_s are the mean free path and albedo of the medium, respectively, and g is the asymmetry parameter of the Henyey-Greenstein phase function [HG40]. The ray query Q defined under the PB coordinate frame includes the incoming position \mathbf{x}_i , incoming direction ω_i , the outgoing position \mathbf{x}_o and outgoing direction ω_o .

Now, the scattering function can be formulated as a ten-dimensional function, conditioned on the PB shape and medium parameters:

$$S(\mathbf{x}_i, \omega_i, \mathbf{x}_o, \omega_o; M, P). \quad (4)$$

As a high-dimensional material function leads to difficulties in representation, we need to reduce the parameter dimensionality. For this, we have two observations. First, the width of the PB is very small compared to the length of the PB, allowing us to discard the position offset along the PB axis. The same assumption has been utilized in previous hair and fur models [MJC*03; YTJR15]. As a result, both the incoming and outgoing positions can be located at the same PB cross-section. Our second observation is that the PB cross-section is a curve established by PB shape parameters. Therefore, the incoming and outgoing positions can be defined with one dimension under the normalized arc-length parameterization and denoted as $c_i, c_o \in [0, 1]$, as illustrated in Fig. 4.

Regarding directions ω_i (ω_o), we transform them into the local space defined by the normal at \mathbf{x}_i (\mathbf{x}_o) and use the parameterization $(h, \phi) \in [0, 1] \times [0, 2\pi]$ in the hemisphere in this local space, where h corresponds to the cosine of the elevation angle and ϕ represents the azimuthal angle. Under this updated parameterization, the scattering function has been simplified from ten to six dimensions:

$$S = S(c_i, h_i, \phi_i, c_o, h_o, \phi_o; M, P). \quad (5)$$

4.2. BPSDF formulation

The scattering function aggregates the scatterings within a Plateau border. Mathematically, it integrates the contribution of all possible light paths that enter the PB at position \mathbf{x}_i with direction ω_i and exit

at position \mathbf{x}_o with direction ω_o :

$$S(c_i, h_i, \phi_i, c_o, h_o, \phi_o; M, P) = \int_{\Omega} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}), \quad (6)$$

where $\bar{\mathbf{x}}$ represents a light path with vertices $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k$ and $\mathbf{x}_0 = \mathbf{x}_i, \mathbf{x}_k = \mathbf{x}_o$. Ω is the path space of entering and exiting the PB from \mathbf{x}_i with ω_i to \mathbf{x}_o with ω_o . f is the contribution of a path $\bar{\mathbf{x}} \in \Omega$.

Among all these possible paths, we identify two types of light paths: one involves only surface reflection or refraction by the PB, and the other involves paths scattered at least once within the PB's medium. This classification is due to the fact that they have distinguished characteristics. Since the surfaces of the PB are liquid films, the first type of paths are all specular, and their distributions are discrete delta functions. The second type of path, influenced by scattering within the medium, has a distribution that takes the form of lobes. Since delta functions are impossible to store or compress, we separate these two types of paths with a specular term S_{specular} and a scattered term $S_{\text{scattered}}$:

$$S(c_i, h_i, \phi_i, c_o, h_o, \phi_o; M, P) = S_{\text{specular}} + S_{\text{scattered}}. \quad (7)$$

Specular term. The reflection and refraction interacting at the PB surface result in various types of specular light paths (Fig. 3), denoted by R, TT, TRT, \dots , where R stands for reflection and T stands for transmission:

$$S_{\text{specular}} = A_{TT} + A_{TRT} + A_{TRRT} + \dots \quad (8)$$

Here, A_t indicates the attenuation along a specular path of type $t \in \{R, TT, TRT, TRRT, \dots\}$, defined as:

$$A_t = \delta(\mathbf{x}_n, \mathbf{x}_o, \omega_o) \times$$

$$(1 - F(\mathbf{x}_0))(1 - F(\mathbf{x}_n))(\prod_{j=1}^{n-1} F(\mathbf{x}_j))e^{-\sigma_t \sum_{j=0}^{n-1} |\mathbf{x}_j \mathbf{x}_{j+1}|}, \quad (9)$$

where $F(\mathbf{x}_j)$ is the Fresnel term at \mathbf{x}_j , $\sigma_t = 1/l$ is the extinction coefficient of medium, and $|\mathbf{x}_j \mathbf{x}_{j+1}|$ is the distance between \mathbf{x}_j and \mathbf{x}_{j+1} . The term $\delta(\mathbf{x}_n, \mathbf{x}_o, \omega_o)$ is the delta function that determines whether \mathbf{x}_o and ω_o correspond to \mathbf{x}_n as the outgoing vertex of a specular path. In practice, we limit the maximum number of bounces to five without noticing any visual difference.

Scattered term. As for the scattered term $f_{\text{scattered}}$, it lacks an analytical solution. Therefore, we design a set of neural networks to represent and importance sample $S_{\text{scattered}}$. The design of these networks is presented in Sec. 5.

5. Neural BPSDF

For the scattered term in the BPSDF, there are two key functions: evaluation and importance sampling. The former means computing the scattered term given incoming and outgoing positions and directions. The latter means, given an incoming position and direction, sample the outgoing position and direction with a probability, which is required by Monte Carlo rendering.

5.1. BPSDF evaluation

We treat the BPSDF evaluation as a regression problem, mapping the input parameters to an output value. To ensure that the neural representation remains efficient and that inference time is mini-

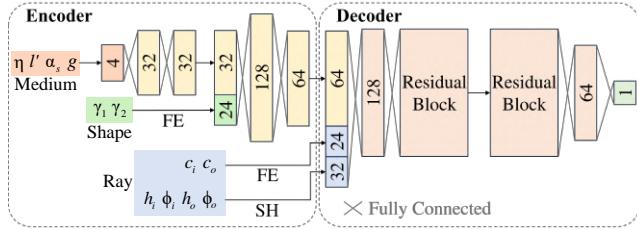


Figure 5: Our Eval-Net consists of an encoder and a decoder, where the former encodes medium and shape parameters into a latent vector, and the latter interprets the latent vector and a ray query into a BPSDF value. The decoder contains two two-layer residual blocks. Frequency encoding (FE) and spherical harmonics (SH) encoding are employed.

mized, we employ two key strategies: reducing the number of input parameters and designing an effective network structure.

For the input parameters, we find that the mfp l and the radius r_p of PB are relative. A PB with a radius r_p containing a medium with a mfp l can be regarded as a PB with a radius 1, where the mfp of the internal medium is l/r_p . Therefore, we introduce a relative mean free path defined as $l' = l/r_p$, which reduces the number of parameters by one. Our evaluation network, which is named Eval-Net, takes the following twelve parameters as inputs:

$$N_{\text{scattered}} = N_{\text{eval}}(\eta, l', \alpha_s, g, \gamma_1, \gamma_2, c_i, h_i, \phi_i, c_o, h_o, \phi_o). \quad (10)$$

Designing a small MLP to directly map twelve parameters to a BPSDF value is a straightforward way. Unfortunately, the reconstructed distribution tends to be inaccurate. To address this issue, we design a neural network with an encoder and a decoder (see Fig. 5). The encoder first extracts the features of the medium parameters as a latent vector by a small MLP. Then, this latent vector is concatenated with the position-encoded shape parameters and fed into a larger MLP to get a feature which represents the liquid and PB shape. Then, the decoder interprets the latent vector along with encoded ray queries into a BPSDF value, where frequency encoding is applied to positions and spherical harmonic (SH) encoding is applied to directions. To improve the network's ability to model sharp distributions, we apply a $e^x - 1$ decoding on the network's output to get the predicted value of Eval-Net. The detailed encodings of inputs are shown in Tab. 2 of the supplementary material. The learned distribution can match the ground truth closely, as shown in Fig. 9 and the supplementary material.

Note that our evaluation network takes single-channel inputs for medium parameters. For RGB-represented l , α_s and g , we independently compute three channels during inference.

5.2. BPSDF importance sampling

To enable BPSDF in Monte Carlo rendering, it's necessary to support importance sampling, indicating that the outgoing position and direction should be sampled, given an incoming position and direction. While the outgoing position and direction are correlated, it's challenging to sample them simultaneously. Therefore, we propose

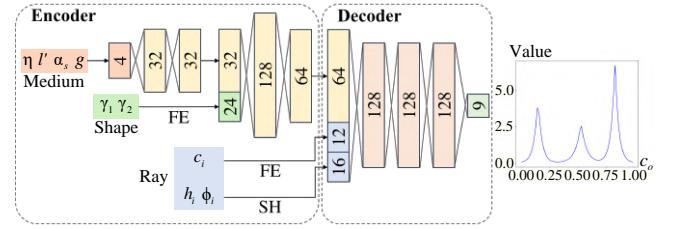


Figure 6: Our Pos-Net has a similar structure as Eval-Net, except the decoder interprets the latent vector, along with the incoming ray parameters, into an outgoing position distribution represented as three Laplace distributions with nine parameters.

using two separate networks: the first network, denoted as Pos-Net, samples the position, and the second network, denoted as Dir-Net, samples the direction, conditioned on the previously sampled position.

Sampling outgoing position. The outgoing position c_o follows a one-dimensional distribution on the PB cross-section. This distribution exhibits a three-lobe shape, which corresponds to the rays that exit from the three faces of the PB (see Fig. 9). Therefore, our key idea is to fit the distribution with three one-dimensional basis functions with a small network, sample one of the basis functions with cumulative distribution function (CDF) sampling, and then sample these functions to obtain an outgoing position.

Specifically, based on the sharp characteristics of outgoing position distributions, we use three Laplace distributions [Ken72] as the basis functions:

$$l_j(x) = \frac{1}{2\lambda_j} e^{-\frac{|x-\mu_j|}{\lambda_j}}, \quad j = 1, 2, 3. \quad (11)$$

Its CDFs are defined as:

$$C_j(x) = \begin{cases} \frac{1}{2} e^{-\frac{\mu_j-x}{\lambda_j}}, & \text{if } x < \mu_j \\ 1 - \frac{1}{2} e^{-\frac{x-\mu_j}{\lambda_j}}, & \text{if } x \geq \mu_j. \end{cases} \quad (12)$$

We use the mixture of three Laplace distributions with weights a_j to fit the distribution of c_o :

$$\text{PDF}_{\text{pos}}(c_o) = \frac{\sum_{j=1}^3 a_j l_j(c_o)}{\sum_{j=1}^3 a_j \int_0^1 l_j(x) dx}. \quad (13)$$

To optimize the value of weights and Laplace distribution parameters, we use a neural network Pos-Net (Fig. 6) to model the mapping:

$$N_{\text{pos}} : (\eta, l', \alpha_s, g, \gamma_1, \gamma_2, c_i, h_i, \phi_i) \rightarrow (\mu_1, \mu_2, \mu_3, \lambda_1, \lambda_2, \lambda_3, a_1, a_2, a_3). \quad (14)$$

Pos-Net has an encoder with the same structure as Eval-Net, and a decoder which maps the extracted feature and (c_i, h_i, ϕ_i) to nine outputs. These outputs are further decoded to obtain parameters of three Laplace distributions. The decoding schemes are shown in Tab. 3 of the supplementary material.

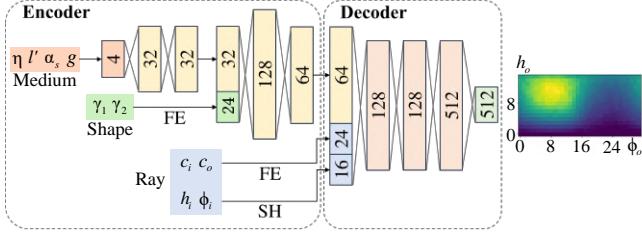


Figure 7: Our Dir-Net has a similar structure as Pos-Net, except the decoder produces the outgoing direction distribution, represented as a 16×32 histogram for (h_o, ϕ_o) conditioned on the outgoing position c_o .

Sampling outgoing direction. With the sampled position c_o , the next step is to sample an outgoing direction ω_o at c_o . As summarized in NeuSample [XWH*23], there are three methods for importance sampling with a neural network: 1) learning a spherical basis function (e.g., SH, or spherical Gaussian) as a proxy and performing analytical sample with the learned proxy; 2) learning the exact PDF directly using normalizing flow or even a diffusion model [FBL*24]; 3) learning a coarse 2D histogram, and performing CDF sampling on the learned histogram. In the context of BPSDF importance sampling, the first solution is not effective, as the BPSDF has a distribution that differs significantly from regular BRDFs, lacking symmetry and regularity. The second method, while capable of producing the most accurate distribution, is time-consuming and can slow down the entire rendering. For these reasons, we opt for the third option, predicting a histogram from a directional sampling network Dir-Net.

Dir-Net takes $(\eta, l', \alpha_s, g, \gamma_1, \gamma_2, c_i, co, h_i, \phi_i)$ as inputs and outputs a histogram with resolution set as 16×32 for $(h_o, \phi_o) \in [0, 1] \times [0, 2\pi]$. As a result, the PDF of (h_o, ϕ_o) is modeled as a piecewise-constant function. The network architecture has a similar encoder as Eval-Net, and then outputs the histogram conditioned on c_i, co, h_i and ϕ_i , as shown in Fig. 7. We also perform the $e^x - 1$ decoding for outputs of Dir-Net. A comparison between the basis function and the histogram representations is shown in Fig. 17.

6. Liquid foam modeling and Monte Carlo rendering

Before using our proposed BPSDF for liquid foam rendering, it is necessary to model a liquid foam with Plateau borders. To our knowledge, no existing approaches can achieve this. Therefore, we propose a practical solution for modeling the liquid foam and then we render it using our BPSDF within a Monte Carlo rendering framework.

6.1. Liquid foam modeling

Existing research [BDWR12] has modeled the structure of bubble collections by simulating bubble flows [QLY*23; BDWR12] using the power diagram, which is also known as the weighted Voronoi diagram [Aur87]. In their approach, each bubble is represented as a sphere, denoted as $B = B(\mathbf{c}, r)$, where \mathbf{c} is the center and r is

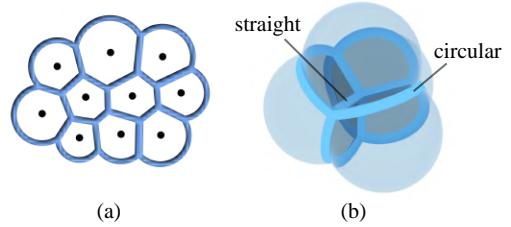


Figure 8: (a) A 2D illustration of the power diagram for bubbles. (b) Examples of several bubbles with PBs generated by a power diagram.

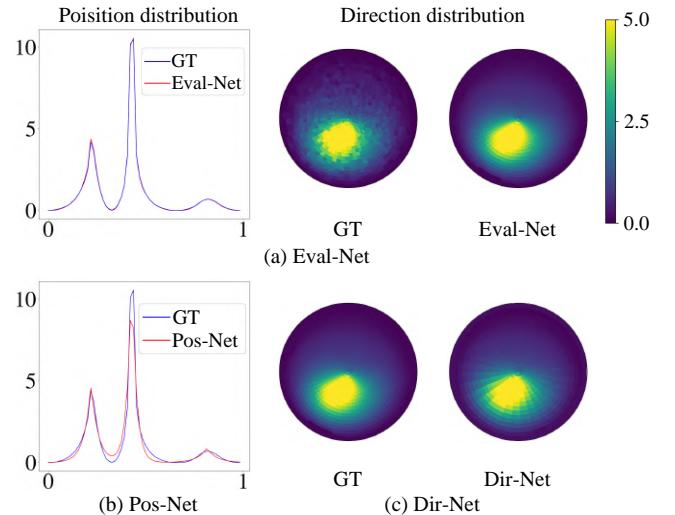


Figure 9: The validation of our Eval-Net (top), Pos-Net (bottom left), and Dir-Net (bottom right), by plotting the BPSDF value as a function of varying positions or directions at a fixed position. All the models can approximately match the GTs. The experiments were conducted under the following settings: ($\eta = 1.33$, $l' = 0.02$, $\alpha_s = 1$, $g = 0.9$, $\gamma_1 = 2.10$, $\gamma_2 = 2.10$, $c_i = 0.23$, $h_i = 0.9$, $\phi_i = 3.14$).

the radius of the bubble. A set of bubbles $\{B_i(\mathbf{c}_i, r_i)\}$ is generated, with \mathbf{c}_i and r_i serving as the center and weight for the corresponding weighted Voronoi cell. The geometry of each bubble is characterized by the intersection of the sphere B_i and its associated Voronoi cell, resulting in a polyhedral shape. The film that forms between two intersecting bubbles is approximated as being planar (see Fig. 8). We follow the same way as prior work, except that we place PBs at the boundaries of the intersection faces where two weighted Voronoi cells meet.

6.2. Monte Carlo rendering of liquid foam

After modeling the geometry of the liquid foam, we treat them as primitives for Monte Carlo rendering. We take path tracing as an example, and integrate our BPSDF into the rendering pipeline. Starting from shooting rays from the camera, if a ray intersects with

a PB with parameters $(\eta, l', \alpha_s, g, \gamma_1, \gamma_2, c_i, h_i, \phi_i)$, we perform the following steps:

- Choose S_{specular} or $S_{\text{scattered}}$. Sample a specular path with its path contribution as the probability. If none of the specular paths is chosen, the scattered term $S_{\text{scattered}}$ is sampled, and preceded to the next step.
- Sample c_o with Pos-Net and compute its PDF: $\text{PDF}_{\text{pos}}(c_o)$.
- Sample ω_o with Dir-Net and compute its PDF: $\text{PDF}_{\text{dir}}(h_o, \phi_o)$.
- Finally, we evaluate the scattered term of BPSDF with Eval-Net to get $S_{\text{scattered}}$ and compute $\frac{S_{\text{scattered}}}{\text{PDF}_{\text{pos}}(c_o)\text{PDF}_{\text{dir}}(h_o, \phi_o)}$ as the sample weights.

The light ray continues bouncing with the new position and direction.

7. Data preparation and training details

In this section, we provide the details of our data generation process and the training setups for our neural networks. More details about the geometric modeling and network architectures can be found in the supplementary.

7.1. Data preparation.

The sampling functions for medium, shape, and ray parameters are detailed in Tab. 2 in the supplementary.

For Eval-Net, we use VPT within a single PB with $r_p = 1$ to generate the dataset. We sample 800 million sets of medium, shapes, and rays. For each sampled parameter set, we compute its BPSDF scattered term in Eqn. (7) with 4096 samples, using 9 hours on an AMD Ryzen 9 5950X CPU with 16 cores and 32 threads.

For Pos-Net, we use the trained Eval-Net to generate the datasets. We sample 2.5 million sets of medium, shape, and incoming ray parameters. For each sampled parameter set, we discretize the outgoing parameters into a $64 \times 16 \times 32$ table and sum over the two-dimensional outgoing direction to get distributions for outgoing positions, using 4 hours on an NVIDIA 4090 GPU.

For Dir-Net, we sample 5 million sets of medium, shape, incoming ray, and outgoing position parameters. We use a 16×32 table to store the outgoing direction distribution for each sampled parameter set, and infer the trained Eval-Net to generate the dataset for Dir-Net, which takes 4 hours on an NVIDIA 4090 GPU.

7.2. Training Details.

We implement our networks in PyTorch and train them on a single NVIDIA 4090 GPU. The Adam optimizer is used with a learning rate of 0.001. For Eval-Net, we use the L_1 loss and train the network for 30 epochs. The learning rate decay is controlled using CosineAnnealingLR, reaching a minimum value of 1×10^{-4} by the end of training. For Pos-Net, we use the Kullback-Leibler Divergence (KL Divergence) as the loss function and train the network for 1000 epochs. For Dir-Net, we use $L_1 + 5L_2$ loss and also train for 1000 epochs. The learning rates of Pos-Net and Dir-Net decay by 0.5 at the 500th epoch. Training three networks takes 24 hours in total.

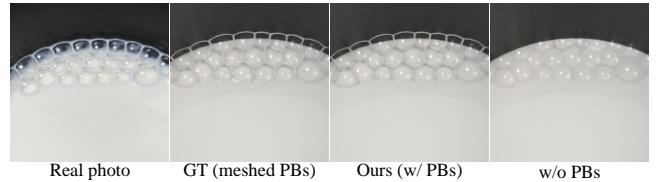


Figure 10: Visual realism caused by modeling PBs in liquid foam. Given a real photo, we create a liquid foam model that matches the real photo as closely as possible and then render it in three ways: with PBs modeled by meshes (GT), with PBs using our model (ours), and without PB modeling, where the latter appears less realistic.

8. Results

We have implemented our BPSDF in unidirectional path tracing inside PBRT-v4 [PJH16], using the WavefrontIntegrator on the GPU. We infer the networks with TensorRT at kFP8 precision. All timings given in this section are measured on a single NVIDIA 3090 GPU. All the reference/ ground truth (GT) images are rendered by volumetric path tracing (VPT) with 2048 samples per pixel (SPP). We compare our method against VPT to demonstrate the effectiveness of our model. We did not compare with other advanced light transport methods, since our model is orthogonal to those techniques. The resolutions of images are set as 1024×1024 . We use the Peak Signal to Noise Ratio (PSNR) to measure the image quality. For fairness in comparison, we only calculate the PSNR of the foam region in the rendered images by using a mask.

8.1. Quality validation

Comparisons with real captured photo. To validate the impact of the Plateau border, we compare the rendered results with and without the modeling of PBs against a real photo in Fig. 10. Additionally, to evaluate the effect of using our PB model, we compare two approaches: meshed modeling and our modeling. We manually use triangle meshes to model PBs to match the real photo as closely as possible, which we take as the GT, while our model employs an approximate PB shape. The comparisons show that incorporating PBs significantly enhances realism, with meshed PBs producing smoother transitions and more natural appearances. Compared to meshed PBs, our modeling incurs only a slight loss in realism while being much easier to apply to large-scale foam. Note that generating a liquid foam to match the real photo is challenging. Therefore, there are still some differences between the real photo and our rendering due to the modeling reasons. Aside from PB modeling, we observe that the films in the photo contain more liquid, while the thin film material we use appears more transparent. While modeling the thin films would affect realism, we consider it parallel to our work, as our primary focus is on PB rendering.

Comparisons with Kück et al. [KVG02]. We compare our method against Kück et al. [KVG02], which models the effect of PBs at the shading level with an empirical shading model instead of at the geometric level. As shown in Fig. 11, our model can easily produce different renderings for a liquid foam containing various types of media by simply adjusting the medium parameters. In contrast,

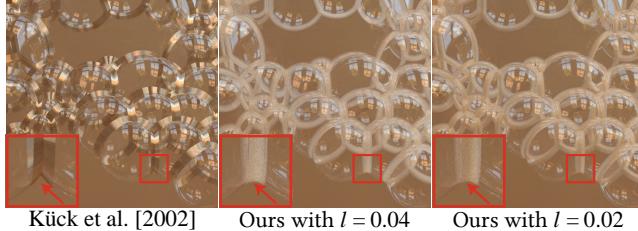


Figure 11: Comparison between Kück et al. [KVG02] and our model. Kück et al.’s method models PBs at the shading level, which lacks the physical parameters to produce different appearances and the PB area is presented on the bubble shell. Our model can render different foams by simply adjusting the medium parameters.

Kück et al.’s method lacks the physical parameters to reproduce these different appearances. Additionally, since their approach does not incorporate PB geometries, the PB area is only represented on the bubble shell, resulting in an unrealistic visual appearance.

Comparisons with volume rendering. To validate the effectiveness of our method, we compare our method against VPT under equal time on two scenes with different media (milk on the top and coffee at the bottom) in Fig. 12, where the coffee has higher absorption than the milk. In both scenes, our method outperforms VPT, showing much less noise visually and qualitatively. The main reason is that the aggregation of scattering modeled by BPSDF shortens the path length and reduces the time cost per sample, leading to more samples under equal-time rendering. By comparing the renderings of two media, our method shows less improvement in the Coffee scene, as the light paths within a medium with higher absorption tend to be shorter. (The average length of light paths is 458 for the Milk scene, and 82 for the Coffee scene.)

We compare our method with VPT across various medium parameters in Fig. 13. We provide renderings of both methods rendered at 2048 SPP, showing that the converged results of our method can match the GTs visually. As expected, for a medium with a shorter mfp and higher albedo, our method shows more improvement, as such media have longer paths. Furthermore, while the mfp and albedo remain the same, our method demonstrates more improvement on medium with a smaller g .

In Fig. 13, we also present the convergence curve for both our method and VPT, by showing the PSNR as a function of rendering time, where the PSNR is computed against the converged renderings (2048 SPP) of VPT. Our method converges rapidly due to lower rendering costs. Note that the PSNR between our method and GT is affected by bias that stems from the network approximation and several assumptions. To investigate the noise level in our renderings further, we also show the PSNR curve between our method and the converged rendering (2048 SPP) of our method, showing that the noise level of our renderings remains lower consistently before and after converging.

White furnace test. In Fig. 14, we perform a white furnace test on a single Plateau border filled with a non-absorptive medium by using a constant environment map with radiance set to 1. As expected,

the specular term leads to dark pixel values. We obtain a constant image by incorporating volume rendering, indicating passing the white furnace test. Our neural BPSDF scattered term together with the specular term produces a nearly constant image, with only slight inaccuracy due to the fitting error of the network.

8.2. Ablation studies

Ablation study for network architecture. The encoder is an important design in all of our networks, and we validate its impact by taking the evaluation network as an example. As shown in Fig. 15, we compare both the rendered results and the radiance distribution with and without the encoder. For fairness, we ensure that the number of parameters in the network (w/o encoder) is the same as in our network. By comparison, we find that the encoder improves the quality of fitting and rendering results.

Ablation study for importance sampling. We validate the effectiveness of two importance sampling networks (Pos-Net and Dir-Net) in Fig. 16 by comparing the results rendered without importance sampling (IS), with Pos-Net only, with Dir-Net only, and with both. While either Pos-Net or Dir-Net reduces the noise, employing both networks leads to the least noise.

Ablation study for Dir-Net. There are several ways to design the Dir-Net, either by predicting the mixture of basis functions or using histograms (our choice). We validate the effectiveness of our choice by comparing it to an alternative method that utilizes eight Spherical Gaussians (SGs) as basis functions. We implement the same network structure as in Fig. 7, with the only difference being that the output 16×32 histogram is replaced with eight SGs. As shown in Fig. 17, we compare rendering and the radiance distribution. Our solution exhibits much less noise than the SG-based solution. Our histogram-based solution has a much closer match to the GT distribution, as the distribution of a BPSDF is asymmetric, which makes it hard for basis functions to fit.

8.3. Limitations and discussion

BPSDF properties. Theoretically, the formulation derived for the BPSDF in Eqn. (4) can ensure both reciprocity and unbiasedness. However, in practice, we make several assumptions which introduces bias. Simultaneously, the neural representation of BPSDF tends to forfeit both reciprocity and unbiasedness, similar to most existing neural materials [FWH^{*}22; ZRW^{*}24]. Despite this limitation, our renderings do not show any noticeable artifacts.

Application of the BPSDF. In this paper, we incorporate our BPSDF into unidirectional path tracing to demonstrate its effectiveness. However, it can also be easily applied to other advanced Monte Carlo rendering techniques, such as volume path guiding [HZE^{*}19]. By leveraging their sampling strategies, we can achieve significantly less noisy renderings.

Difference from BCSDF. Our BPSDF, which is connected to PB, differs from the BCSDF used in previous hair and fur models in several aspects. In terms of assumptions, our BPSDF discards the

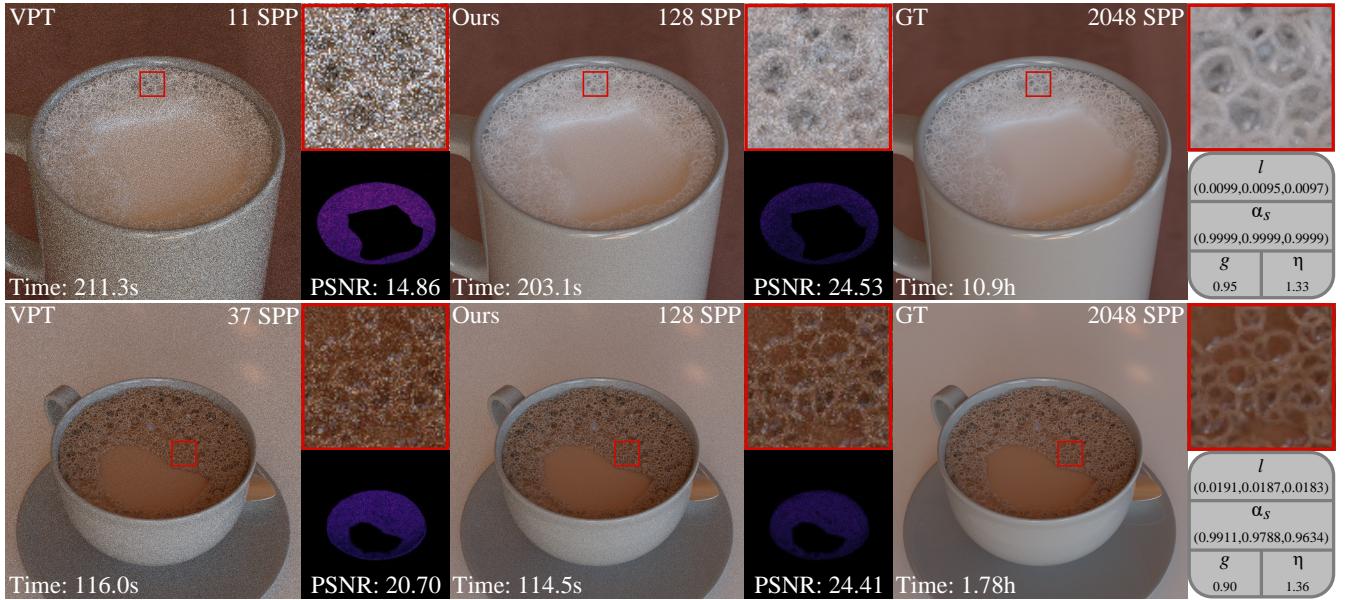


Figure 12: Comparisons between our method and VPT on two different scenes. For mediums with small l and large α_s , our method significantly reduces rendering time overhead, while VPT suffers from multiple scattering in PBs.

offset along the PB axis, while the BCSDF assumes that the outgoing position remains unchanged. For formulation and computation, our BPSDF involves more parameters and aggregates the scattered paths in a distribution represented by networks, while BCSDF divides scattered paths into several types and utilizes precomputed tables. In Fig. 18, we compare the bias introduced by the assumptions of our BPSDF and the BCSDF. We apply assumptions to VPT to ignore either the longitudinal offset or the outgoing position displacement of light propagation within the PB. We find that assuming a fixed outgoing position introduces significant error, while neglecting longitudinal offset results in minimal bias. This suggests that the rendering result is more sensitive to the outgoing position on the PB cross-section, and validates the reasonableness of our assumptions.

Difference from BSSRDF. The BPSDF with full parameters shares the same parameterization as the BSSRDF, including both the incident and exit positions and directions. However, in the context of the PB, it has a distinctive geometry, which leads to a different ray parameterization and a specialized scattering distribution. Specifically, our simplified BPSDF parameterizes both incident and exit positions on the PB cross-section, resulting in characteristic positional and directional distributions. Compared to existing neural BSSRDF approaches such as Neural SSS [TTJ*24], our approach uses specialized network designs to handle the distributions of BPSDF. For example, we adopt three Laplace distributions to sample the outgoing position and a histogram-based method for the asymmetric directional distribution. In contrast, Neural SSS uses a 3D Gaussian to fit simpler distributions for importance sampling.

Generalization to complex PB shapes. We model the PB geometry by making several practical assumptions, such as maintaining a

constant radius along the Plateau axis and ignoring the PB’s junctions. Our model also lacks a simulation of how different liquids affect the size of the PB. More accurate PB geometries, such as a bone-like shape, can be employed, as illustrated in Fig. 1 of the supplementary material. Our approach can be generalized to such PB shapes by modifying the networks. For example, additional parameters for PB shape and longitudinal positions need to be incorporated, which is feasible but increases the complexity of both networks and importance sampling. We leave it for future work.

9. Conclusion

In this paper, we have presented a novel way for liquid foam rendering, which can produce realistic renderings efficiently. The key idea is to explicitly model the Plateau border into liquid foam and aggregate the scatterings within a PB using a newly proposed scattering function known as BPSDF. We have developed a comprehensive set of neural functions for representing the BPSDF and for importance sampling to facilitate practical applications. As a result, our method allows for the realistic rendering of liquid foam, while the BPSDF demonstrates a significant reduction in noise compared to direct volumetric rendering.

We believe that the modeling of the PB is novel to graphics and has the potential to open up new possibilities in related areas. In the future, it will be interesting to introduce PB into foam simulation. Furthermore, aggregating the scatterings in a collection of PBs rather than in an individual PB can further reduce noise levels.

Acknowledgments

We thank the reviewers for the valuable comments. This work is supported by Laoshan Laboratory (No. LSKJ202300305) and the

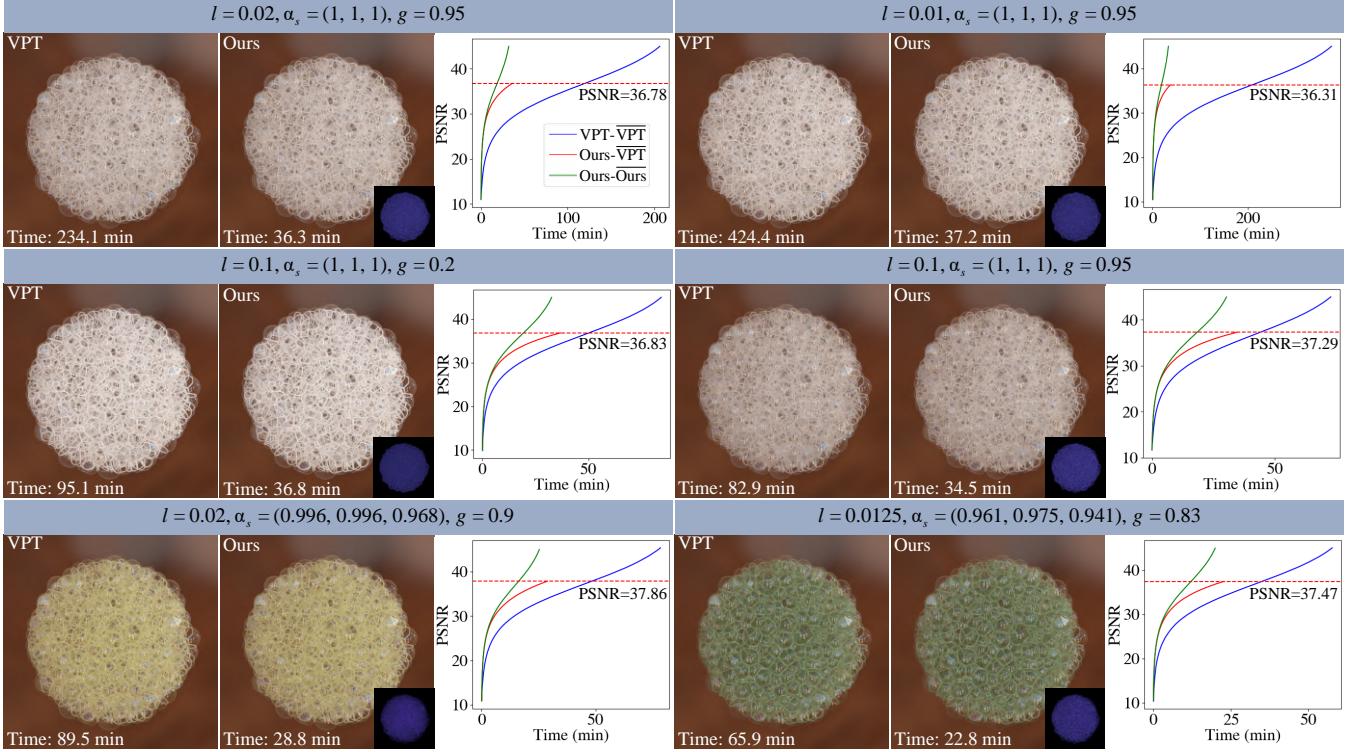


Figure 13: Comparisons between our method and VPT on varying medium parameters. In the convergence curves, we use \overline{VPT} and \overline{Ours} to represent the converged results (2048 SPP) of VPT and ours. Our method leads to a fast convergence but introduces bias compared to VPT as illustrated by the convergence curves. η is set to 1.33. The error images are scaled by a factor of 3.

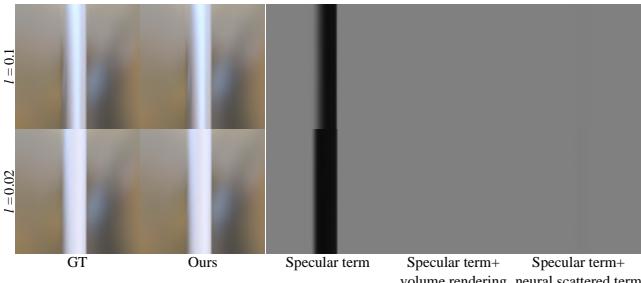


Figure 14: A white furnace test on a single PB for our BPSDF with no absorption ($\alpha_s = 1$). The specular term results in darker pixel values. Including the scattered term of the BPSDF using volume rendering produces a constant image. Including the neural scattered term for the BPSDF closely approximates a constant image, with some minor inaccuracies due to network bias.

National Natural Science Foundation of China (No. 62025207 and 62172220). Beibei Wang and Ligang Liu are the corresponding authors.

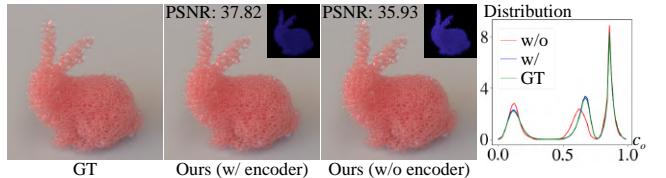


Figure 15: The impact of the encoder in our evaluation network. The encoder improves the accuracy of the rendered result (2048 SPP) and produces a distribution that is closer to the GT, which is computed using Monte Carlo sampling. The error images are scaled by a factor of 3.

References

- [Aur87] AURENHAMMER, F. “Power Diagrams: Properties, Algorithms and Applications”. *SIAM Journal on Computing* 16.1 (1987), 78–96 6.
- [BDWR12] BUSARYEV, OLEKSIY, DEY, TAMAL K., WANG, HUAMIN, and REN, ZHONG. “Animating bubble interactions in a liquid foam”. *ACM Trans. Graph.* 31.4 (July 2012). ISSN: 0730-0301 6.
- [BKD*08] BARRETT, D., KELLY, SEAN, DALY, E., et al. “Taking Plateau into Microgravity: The Formation of an Eightfold Vertex in a System of Soap Films”. *Microgravity Science and Technology* 20 (June 2008), 17–22 2, 3.
- [Bra92] BRAKKE, KENNETH A. “The Surface Evolver”. *Experimental Mathematics* 1.2 (1992), 141–165 2.

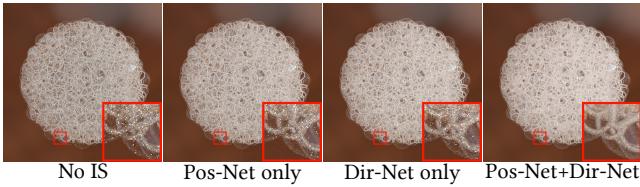


Figure 16: Comparison of renderings with equal SPP across different sampling strategies. Our choice, utilizing both position and direction importance sampling, results in the lowest level of noise.

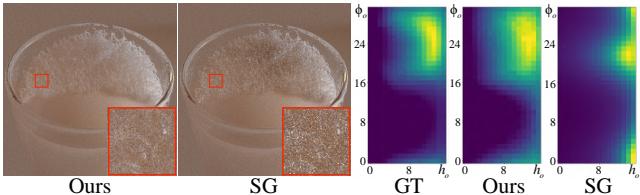


Figure 17: Rendering and distribution comparison using an SG mixture (eight SGs) and a histogram (ours) for neural directional IS, where the GTs are computed with Monte Carlo sampling. Our method demonstrates less noise and achieves a distribution that more closely matches the GT distributions.



Figure 18: The impact of assumptions in our BPSDF and the BCSDF. Our BPSDF ignores the longitudinal offset, causing negligible error, while BCSDF assumes the outgoing position remains unchanged, leading to a more considerable error. The error images are scaled by a factor of 3.

- [BSW10] BAGAR, FLORIAN, SCHERZER, DANIEL, and WIMMER, MICHAEL. “A Layered Particle-Based Fluid Model for Real-Time Rendering of Water”. *Computer Graphics Forum* 29.4 (2010), 1383–1389 2.
- [CCE*13] CANTAT, ISABELLE, COHEN-ADDA, SYLVIE, ELIAS, FLORENCE, et al. *Foams: Structure and Dynamics*. Oxford University Press, July 2013. ISBN: 9780199662890 2.
- [CWW24] CHEN, XIANG, WANG, LU, and WANG, BEIBEI. “Real-time Neural Woven Fabric Rendering”. *ACM SIGGRAPH 2024 Conference Papers*. SIGGRAPH ’24. Denver, CO, USA: Association for Computing Machinery, 2024 3.
- [DCGG11] DARLES, E., CRESPIN, B., GHAZANFARPOUR, D., and GONZATO, J.C. “A Survey of Ocean Simulation and Rendering Techniques in Computer Graphics”. *Computer Graphics Forum* 30.1 (2011), 43–60 2.
- [DHV*23] DROSKE, MARC, HANIKA, JOHANNES, VORBA, JÍŘI, et al. “Path tracing in Production: The Path of Water”. *ACM SIGGRAPH 2023 Courses*. SIGGRAPH ’23. Los Angeles, California: Association for Computing Machinery, 2023 2.
- [DWK*22] DENG, YITONG, WANG, MENGDI, KONG, XIANGXIN, et al. “A moving eulerian-lagrangian particle method for thin film and foam simulation”. *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301 2.
- [FBL*24] FU, ZIYANG, BELHE, YASH, LU, HAOLIN, et al. “BSDF importance sampling using a diffusion model”. *SIGGRAPH Asia 2024 Conference Papers*. SA ’24. Association for Computing Machinery, 2024 3, 6.
- [FR22] FISCHER, MICHAEL and RITSCHEL, TOBIAS. “Metapearance: Meta-Learning for Visual Appearance Reproduction”. *ACM Trans. Graph.* 41.6 (Nov. 2022). ISSN: 0730-0301 3.
- [FWH*22] FAN, JIAHUI, WANG, BEIBEI, HASAN, MILOS, et al. “Neural Layered BRDFs”. *ACM SIGGRAPH 2022 Conference Proceedings*. SIGGRAPH ’22. Vancouver, BC, Canada: Association for Computing Machinery, 2022. ISBN: 9781450393379 3, 8.
- [FWH*23] FAN, JIAHUI, WANG, BEIBEI, HASAN, MILOS, et al. “Neural Biplane Representation for BTF Rendering and Acquisition”. *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH ’23. Los Angeles, CA, USA: Association for Computing Machinery, 2023 3.
- [GH04] GREENWOOD, S. T. and HOUSE, D. H. “Better with bubbles: enhancing the visual realism of simulated fluid”. *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA ’04. Grenoble, France: Eurographics Association, 2004, 287–296. ISBN: 3905673142 2, 3.
- [GLH*23] GUO, JIE, LI, ZERU, HE, XUEYAN, et al. “MetaLayer: A Meta-Learned BSDF Model for Layered Materials”. *ACM Trans. Graph.* 42.6 (Dec. 2023). ISSN: 0730-0301 3.
- [GSP*19] GIL, SANGHYEON, SEOK, YUNJI, PARK, KIYEOL, et al. “Soap film flow and thickness for soap bubble rendering”. *Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology*. VRST ’19. Parramatta, NSW, Australia: Association for Computing Machinery, 2019. ISBN: 9781450370011 2.
- [GZC004] GÉMINARD, J. C., ŹYWOCINSKI, A., CAILLIER, F., and OSWALD, P. “Observation of negative line tensions from Plateau border regions in dry foam films”. *Philosophical Magazine Letters* 84.3 (2004), 199–204 2.
- [HG40] HENHEY, L. C. and GREENSTEIN, JESSE LEONARD. “Diffuse radiation in the Galaxy”. *The Astrophysical Journal* 93 (1940), 70–83 4.
- [HGC*20] HU, BINGYANG, GUO, JIE, CHEN, YANJUN, et al. “DeepBRDF: A Deep Representation for Manipulating Measured BRDF”. *Computer Graphics Forum* 39.2 (2020), 157–166 3.
- [HIK*20] HUANG, WEIZHEN, ISERINGHAUSEN, JULIAN, KNEIPHOF, TOM, et al. “Chemomechanical simulation of soap film flow on spherical bubbles”. *ACM Trans. Graph.* 39.4 (Aug. 2020). ISSN: 0730-0301 2.
- [HYL*23] HU, JINKAI, YU, CHENGZHONG, LIU, HONGLI, et al. “Deep Real-time Volumetric Rendering Using Multi-feature Fusion”. *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH ’23. Los Angeles, CA, USA: Association for Computing Machinery, 2023 3.
- [HZE*19] HERHOLZ, SEBASTIAN, ZHAO, YANGYANG, ELEK, OSKAR, et al. “Volume Path Guiding Based on Zero-Variance Random Walk Theory”. *ACM Trans. Graph.* 38.3 (June 2019). ISSN: 0730-0301 8.
- [IMN04] IWASAKI, KEI, MATSUZAWA, KEICHI, and NISHITA, TOMOYUKI. “Real-Time Rendering of Soap Bubbles Taking into Account Light Interference”. *Proceedings of the Computer Graphics International*. CGI ’04. USA: IEEE Computer Society, 2004, 344–348 2.
- [ISN*20] ISHIDA, SADASHIGE, SYNAK, PETER, NARITA, FUMIYA, et al. “A model for soap film dynamics with evolving thickness”. *ACM Trans. Graph.* 39.4 (Aug. 2020). ISSN: 0730-0301 2.
- [IYAH17] ISHIDA, SADASHIGE, YAMAMOTO, MASAFUMI, ANDO, RYOICHI, and HACHISUKA, TOSHIYA. “A hyperbolic geometric flow for evolving films and foams”. *ACM Trans. Graph.* 36.6 (Nov. 2017). ISSN: 0730-0301 2.
- [Ken72] KENDALL, DAVID. “An Introduction to Probability Theory and its Applications, Vol. II. By William Feller. 2nd Edition. 1971. (John Wiley and Sons.)” *The Mathematical Gazette* 56.395 (1972), 65–66 5.

- [KMM*17] KALLWEIT, SIMON, MÜLLER, THOMAS, MCWILLIAMS, BRIAN, et al. “Deep scattering: rendering atmospheric clouds with radiance-predicting neural networks”. *ACM Trans. Graph.* 36.6 (Nov. 2017). ISSN: 0730-0301 3.
- [KMX*21] KUZNETSOV, ALEXANDR, MULLIA, KRISHNA, XU, ZEXI-ANG, et al. “NeuMIP: multi-resolution neural materials”. *ACM Trans. Graph.* 40.4 (July 2021). ISSN: 0730-0301 3.
- [KR87] KO CZÓ, K. and RÁCZ, G. “Flow in a plateau border”. *Colloids and Surfaces* 22.2 (1987), 95–96. ISSN: 0166-6622 2, 3.
- [KVG02] KÜCK, HENDRIK, VOGELSGANG, CHRISTIAN, and GREINER, GÜNTHER. “Simulation and Rendering of Liquid Foams”. *Graphics Interface*. 2002 2, 3, 7, 8.
- [KWM*22a] KUZNETSOV, ALEXANDR, WANG, XUEZHENG, MULLIA, KRISHNA, et al. “Rendering Neural Materials on Curved Surfaces”. *ACM SIGGRAPH 2022 Conference Proceedings*. SIGGRAPH ’22. Vancouver, BC, Canada: Association for Computing Machinery, 2022. ISBN: 9781450393379 3.
- [KWM*22b] KUZNETSOV, ALEXANDR, WANG, XUEZHENG, MULLIA, KRISHNA, et al. “Rendering Neural Materials on Curved Surfaces”. *ACM SIGGRAPH 2022 Conference Proceedings*. SIGGRAPH ’22. Vancouver, BC, Canada: Association for Computing Machinery, 2022. ISBN: 9781450393379 3.
- [LGH*24] LIANG, SHIYU, GAO, YANG, HU, CHONGHAO, et al. “State of the Art in Efficient Translucent Material Rendering with BSSRDF”. *Computer Graphics Forum* 43.1 (2024), e14998 3.
- [LHL*24] LITALIEN, JOEY, HAŠAN, MILOŠ, LUAN, FUJUN, et al. “Neural Product Importance Sampling via Warp Composition”. *SIGGRAPH Asia 2024 Conference Papers*. SA ’24. Association for Computing Machinery, 2024 3.
- [LHW21] LEONARD, L., HÖHLEIN, K., and WESTERMANN, R. “Learning Multiple-Scattering Solutions for Sphere-Tracing of Volumetric Subsurface Effects”. *Computer Graphics Forum* 40.2 (2021), 165–178 3.
- [LL65] LEONARD, RALPH A. and LEMLICH, ROBERT. “A study of interstitial liquid flow in foam. Part I. Theoretical model and application to foam fractionation”. *AIChe Journal* 11.1 (1965), 18–25 2, 3.
- [MJC*03] MARSCHNER, STEPHEN R., JENSEN, HENRIK WANN, CAM-MARANO, MIKE, et al. “Light scattering from human hair fibers”. *ACM Trans. Graph.* 22.3 (July 2003), 780–791. ISSN: 0730-0301 3, 4.
- [Mor03] MORRIS, NIGEL J. W. “Capturing the Reflectance Model of Soap Bubbles”. 2003 2.
- [Ngu02] NGUYEN, ANH V. “Liquid Drainage in Single Plateau Borders of Foam”. *Journal of Colloid and Interface Science* 249.1 (2002), 194–199. ISSN: 0021-9797 2.
- [PJH16] PHARR, MATT, JAKOB, WENZEL, and HUMPHREYS, GREG. *Physically Based Rendering: From Theory to Implementation* (3rd ed.). 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Oct. 2016, 1266. ISBN: 9780128006450 7.
- [Pla73] PLATEAU, JOSEPH ANTOINE FERDINAND. *Statique expérimentale et théorique des liquides soumis aux seules forces moléculaires*. Vol. 2. Gauthier-Villars, 1873 2.
- [QLY*23] QU, ZIYIN, LI, MINCHEN, YANG, YIN, et al. “Power Plastics: A Hybrid Lagrangian/Eulerian Solver for Mesoscale Inelastic Flows”. *ACM Trans. Graph.* 42.6 (Dec. 2023). ISSN: 0730-0301 2, 6.
- [RGW20] RAINER, GILLES, GHOSH, ABHIJEET, JAKOB, WENZEL, and WEYRICH, TIM. “Unified Neural Encoding of BTFS”. *Computer Graphics Forum (Proc. Eurographics)* 39.2 (July 2020), 167–178 3.
- [RJG19] RAINER, GILLES, JAKOB, WENZEL, GHOSH, ABHIJEET, and WEYRICH, TIM. “Neural BTFS Compression and Interpolation”. *Computer Graphics Forum (Proceedings of Eurographics)* 38.2 (Mar. 2019) 3.
- [SRRW21] SZTRAJMAN, ALEJANDRO, RAINER, GILLES, RITSCHEL, TOBIAS, and WEYRICH, TIM. “Neural BRDF Representation and Importance Sampling”. *Computer Graphics Forum* 40.6 (2021), 332–346 3.
- [STW08] SUN, QICHENG, TAN, LU, and WANG, GUANGQIAN. “LIQUID FOAM DRAINAGE: AN OVERVIEW”. *International Journal of Modern Physics B* 22.15 (2008), 2333–2354 2.
- [TFRJ24] TG, T., FRISVAD, J. R., RAMAMOORTHI, R., and JENSEN, H. W. “NeuPreSS: Compact Neural Precomputed Subsurface Scattering for Distant Lighting of Heterogeneous Translucent Objects”. *Computer Graphics Forum* 43.7 (2024), e15234 3.
- [TTJ*24] TG, T., TRAN, D. M., JENSEN, H. W., et al. “Neural SSS: Lightweight Object Appearance Representation”. *Computer Graphics Forum* 43.4 (2024), e15158 3, 9.
- [TUGM22] TONGBUASIRILAI, TANABOON, UNGER, JONAS, GUILLEMOT, CHRISTINE, and MIANDJI, EHSAN. “A Sparse Non-parametric BRDF Model”. *ACM Trans. Graph.* 41.5 (Oct. 2022). ISSN: 0730-0301 3.
- [vdLGS09] VAN DER LAAN, VLADIMIR J., GREEN, SIMON, and SAINZ, MIGUEL. “Screen space fluid rendering with curvature flow”. *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*. I3D ’09. Boston, Massachusetts: Association for Computing Machinery, 2009, 91–98. ISBN: 9781605584294 2.
- [VKJ19] VICINI, DELIO, KOLTUN, VLADLEN, and JAKOB, WENZEL. “A learned shape-adaptive subsurface scattering model”. *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301 3.
- [WFS22] WRETBORN, JOEL, FLYNN, SEAN, and STOMAKHIN, ALEXEY. “Guided bubbles and wet foam for realistic whitewater simulation”. *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301 2.
- [WH00] WEAIRE, DENIS and HÜTZLER, STEFAN. *The Physics of Foams*. Oxford University Press, Jan. 2000. ISBN: 9780198505518 2.
- [WJHY22] WANG, BEIBEI, JIN, WENHUA, HAŠAN, MILOŠ, and YAN, LING-QI. “SpongeCake: A Layered Microflake Surface Appearance Model”. *ACM Trans. Graph.* 42.1 (Sept. 2022). ISSN: 0730-0301 3.
- [XMWY24] XU, ZILIN, MONTAZERI, ZAHRA, WANG, BEIBEI, and YAN, LING-QI. “A Dynamic By-example BTF Synthesis Scheme”. *SIGGRAPH Asia 2024 Conference Papers*. SA ’24. Association for Computing Machinery, 2024 3.
- [XWH*23] XU, BING, WU, LIWEN, HASAN, MILOS, et al. “NeuSample: Importance Sampling for Neural Materials”. *ACM SIGGRAPH 2023 Conference Proceedings*. SIGGRAPH ’23. Los Angeles, CA, USA: Association for Computing Machinery, 2023 3, 6.
- [XWPB17] XU, JIE, WU, TAO, PENG, CHUANG, and ADEGBITE, STEPHEN. “Geometric modeling of Plateau borders using the orthographic projection method for closed cell rigid polyurethane foam thermal conductivity prediction”. *Modelling and Simulation in Materials Science and Engineering* 25.6 (Aug. 2017), 065021 2.
- [XZJM24] XUE, BOWEN, ZHAO, SHUANG, JENSEN, HENRIK WANN, and MONTAZERI, ZAHRA. *A Hierarchical Architecture for Neural Materials*. 2024 3.
- [YTJR15] YAN, LING-QI, TSENG, CHI-WEI, JENSEN, HENRIK WANN, and RAMAMOORTHI, RAVI. “Physically-accurate fur reflectance: modeling, measurement and rendering”. *ACM Trans. Graph.* 34.6 (Nov. 2015). ISSN: 0730-0301 4.
- [ZK24] ZINGSHEIM, DOMENIC and KLEIN, REINHARD. “Learning subsurface scattering solutions of tightly-packed granular media using optimal transport”. *Computers & Graphics* 119 (2024), 103895. ISSN: 0097-8493 3.
- [ZRW*24] ZELTNER, TIZIAN, ROUSSELLE, FABRICE, WEIDLICH, ANDREA, et al. “Real-time Neural Appearance Models”. *ACM Transactions on Graphics* 43.3 (June 2024), 1–17. ISSN: 1557-7368 3, 8.
- [ZZW*21] ZHENG, CHUANKUN, ZHENG, Ruzhang, WANG, RUI, et al. “A Compact Representation of Measured BRDFs Using Neural Processes”. *ACM Trans. Graph.* 41.2 (Nov. 2021). ISSN: 0730-0301 3.
- [ZZW*22] ZHU, JUNQIU, ZHAO, SIZHE, WANG, LU, et al. “Practical level-of-detail aggregation of fur appearance”. *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301 3.