UltraRay: Introducing Full-Path Ray Tracing in Physics-Based Ultrasound Simulation

Felix Duelmer[1,2,3,4⋆], Mohammad Farid Azampour[1,2], Magdalena Wysocki[1,2], and Nassir Navab[1,2]

[1] Chair for Computer Aided Medical Procedures (CAMP), Technical University of Munich, Germany
[2] Munich Center for Machine Learning (MCML), Munich, Germany
[3] Institute of Biological and Medical Imaging, Bioengineering Center, Helmholtz Zentrum München, Neuherberg, Germany
[4] Chair of Biological Imaging, Central Institute for Translational Cancer Research (TranslaTUM), School of Medicine and Health & School of Computation, Information and Technology, Technical University of Munich, Munich, Germany

Abstract. Traditional ultrasound simulators solve the wave equation to model pressure distribution fields, achieving physical accuracy but requiring significant computational time and resources. Ray tracing approaches have been introduced to address this limitation, modeling wave propagation as rays interacting with boundaries and scatterers. However, existing models simplify ray propagation, generating echoes at interaction points without considering return paths to the sensor. This can result in undesired artifacts and necessitates careful scene tuning for plausible results. We propose UltraRay, a novel framework that models the full path of acoustic waves reflecting from tissue boundaries. We derive the equations for accurate reflection modeling across multiple interaction points and introduce a sampling strategy for an increased likelihood of a ray returning to the transducer. By incorporating a ray emission scheme for plane wave imaging and a standard signal processing pipeline for beamforming, we are able to simulate the ultrasound image formation process end-to-end. Built on a differentiable modular framework, UltraRay introduces an extendable foundation for differentiable ultrasound simulation based on full-path ray tracing. We demonstrate its advantages compared to the state-of-the-art ray tracing ultrasound simulation, shown both on a synthetic scene and a spine phantom.

Keywords: Ultrasound · Physics-Based Simulation · Reflection Modeling · Ray Tracing.

1 Introduction

Ultrasound simulation is important in medical imaging, with applications in areas like training sonographers [7, 16], designing and testing transducers [5, 6, 11], or, more recently, generating data for training neural networks [25, 2, 1].

⋆ felix.duelmer@tum.de

Traditional ultrasound simulators solve the wave equation using Green's function, either with linear methods [13] or nonlinear approaches [24]. While these methods are accurate, they are computationally intensive, as they require solving complex mathematical equations over fine spatial and temporal grids. This makes them slow and difficult to apply for real-time applications [21], large-scale simulations [4], or waveform inversion tasks [17]. Ray tracing algorithms were introduced to address these limitations [18, 3]. Instead of simulating entire wavefields, these algorithms trace individual rays through a pre-defined model of tissue structures and parameters, allowing the computational workload to be efficiently parallelized. Further enhancements in simulation realism have been achieved by integrating improved scattering models [10] and utilizing Monte Carlo ray tracing (MCRT) techniques [18, 1].

However, current ray tracing methods typically only account for the ray traveling from the transducer to the scattering or reflection event, directly adding an echo to the time-echo signal without verifying whether the ray can be received at the sensor. Therefore, these simulators typically don't include beamforming algorithms because they assume the ray-tracing data already reflects beamformed channel information. This simplification can lead to unrealistic reflections, introducing implausible artifacts in the synthesized image, which can, e.g., impact neural network training.

Similar challenges arise in the visible light domain as in ultrasound, including phenomena like reflections, refractions, and scattering. To address these, physically based renderers [22] have been developed to create photorealistic images by explicitly simulating the complete light transport process using ray tracing. Examples of such fast and accurate simulators include Mitsuba 3 [12], which not only delivers photorealistic forward simulations but is also built with automatic differentiation, making the simulation process invertible and thus useful for parameter or shape optimization [15, 19, 28].

Beyond achieving accurate forward modeling of image formation, the ability to invert the process has drawn interest from the ultrasound community. Recent approaches, such as UltraNeRF [27], combine a fast ultrasound simulation with implicit neural representations to facilitate efficient ultrasound image synthesis. However, by using a simplified image formation model, e.g. not considering secondary effects such as reflections, they restrict their capability to fully represent the complexities inherent in ultrasound physics.

To address this gap, we introduce UltraRay, a framework for fast ultrasound reflection simulation based on ray tracing (see Figure 1). Our key contributions are as follows:

- We propose a full-path ray tracing method that tracks rays from emission all the way to reception. By introducing a transducer sampling strategy, we present an efficient way of returning the rays to the transducer after a scene interaction.
- We derive the mathematical formulation for our Monte Carlo-based ray tracing method and integrate a physics-informed model for tissue boundary reflections.

– By incorporating a plane wave imaging acquisition scheme and a conventional beamforming pipeline - including delay-and-sum beamforming, envelope detection, and log compression - we enable end-to-end ultrasound simulation.

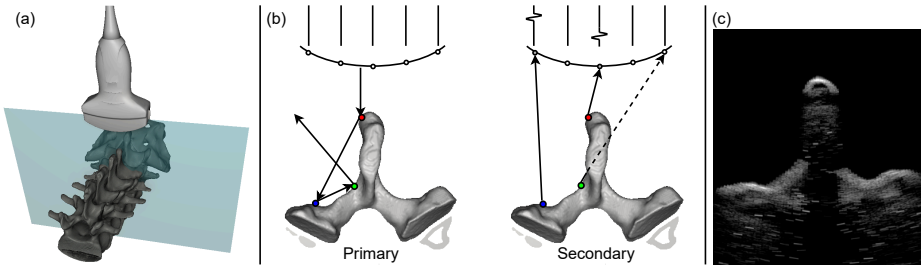

Fig. 1. Visualization of the simulation pipeline. a) We start be defining the virtual scene, which includes the transducer and acquisition plane. b) Exemplary demonstration of a single ray traversing the scene. In the primary phase, a ray is emitted from the transducer and traverses the scene. At each interaction point with the vertebrae, secondary rays are cast toward the center of a randomly sampled transducer element. If a secondary ray is not blocked, its contribution is added to the corresponding time-pressure signal of that transducer element. In (c), we present the resulting B-mode image generated by the proposed method.

2 Methodology

2.1 Ray Tracing in Ultrasound

In ultrasound imaging, our goal is to find the pressure signal $P(e,t)$ at a transducer element e over time t. By modeling incoming pressure waves as rays, the contribution to an element can be written as:

$$P(e,t) = \int_{\Omega}\int_{A} P_i(\mathbf{x},t,\omega_i)\, f_d(\omega_i)\, d\omega\, da, \tag{1}$$

where ω_i is the direction of a ray over the hemisphere Ω, originating from point $\mathbf{x}$. The integral over the transducer surface A accounts for the spatial extent of the element. P_i represents the incoming pressure intensity, and f_d is the directivity function, which weighs the contribution of pressure from each direction, accounting for the directional sensitivity of the transducer element.

To analyze the pressure originating from a point $\mathbf{x}$, we adapt the general rendering equation [14] to ultrasound in order to evaluate the incident P_i, outgoing P_o, and emitted pressure P_e at that location:

$$P_o(\mathbf{x},t,\omega_o) = P_e(\mathbf{x},t,\omega_o) + \int_{\Omega} f_r(\mathbf{x},\omega_i,\omega_o)\, P_i(\mathbf{x},t,\omega_i)\, (\mathbf{n}\cdot\omega_i)\, d\omega_i, \tag{2}$$

ω_o and ω_i denote the incoming and outgoing directions, and $\mathbf{n}$ is the surface normal. Each ray is weighted by a bidirectional scattering distribution function (BSDF) f_r and a geometric term $(\mathbf{n} \cdot \omega_i)$ where $\cdot$ represents the dot product. In contrast to Amadou *et al.* [1], we explicitly include the emitter pressure P_e modeling a complete round-trip. Because directly computing these integrals is computationally infeasible, we use a MCRT strategy to efficiently aggregate ray contributions.

2.2 Monte-Carlo Ray tracing

Following [1, 18], we model ultrasound simulation using MCRT. Rewriting $P(e,t)$ to include a stochastic sampling process, we derive:

$$P(e,t) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{P_i(\mathbf{x}, t, \omega_i)\, f_d(\omega_i)}{p_t(\omega_i, a)}, \tag{3}$$

where the integrals are replaced by sampling directions ω_i and locations a on the transducer surface, weighted by the directivity function $f_d(\omega_i)$ and the inverse of its sampling probability $p_t(\omega_i, a)$. The result is averaged over N rays.

Similarly, the equilibrium of incoming and outgoing rays at a surface point can be rewritten to:

$$P_o(\mathbf{x}, t, \omega_o) \approx P_e(\mathbf{x}, t, \omega_o) + \frac{1}{N} \sum_{i=1}^{N} \frac{f_d(\mathbf{x}, \omega_i, \omega_o) P_i(\mathbf{x}, t, \omega_i)(\mathbf{n} \cdot \omega_i)}{p(\omega_i)} \tag{4}$$

where the sampled direction ω_i is again weighted by the inverse of its sampling probability $p(\omega_i)$, which accounts for the likelihood of sampling this direction based on the geometrical terms and the BSDF.

2.3 Proposed approach

Ray Emission In an ultrasound acquisition, acoustic emission of specific elements is delayed to form focused, diverging, or planar wavefronts. We adopt this approach for our plane wave imaging , adding delays based on the transmission angle. Since rays are independent, we can emit rays with various delays for multiple angles simultaneously without them interfering in the scene. To account for transducer directivity, we include a weighting function f_d in the outgoing pressure $P_o(\omega_o)$. Each ray is normalized by the number of rays N and multiplied by a cosine term:

$$f_d(\omega_o) = \frac{1}{N} (\omega_o \cdot \mathbf{n}), \tag{5}$$

where ω_o is the outgoing direction and $\mathbf{n}$ is the transducer surface normal.

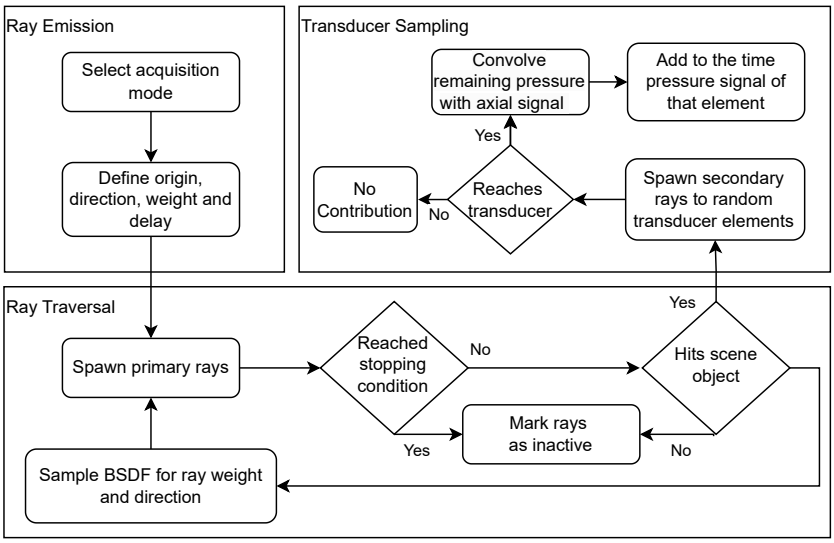

Fig. 2. Flow diagram on the main parts of the ray tracing: the ray emission based on the transducer geometry, the ray traversal through the scene, and the transducer sampling to enhance the likelihood of the rays reaching the transducer

Ray Traversal We follow each primary ray (reflection/refraction in the medium) until it misses all objects or exceeds a bounce limit. At each intersection, we trigger a transducer sampling step (see next section) and then compute the new direction and amplitude via the BSDF (see Figure 2). Similar to previous works [18, 1], we model reflection and transmission at tissue boundaries depending on the acoustic impedance ratio $\eta = Z_1/Z_2$. From the incident direction ω_i and surface normal $\mathbf{n}$, we can calculate the reflected and transmitted direction using Snell's law to:

$$\omega_r = \omega_i + 2\cos\theta_r\,\mathbf{n}, \quad \omega_t = \eta\,\omega_i + (\eta\cos\theta_r - \cos\theta_t)\,\mathbf{n}. \tag{6}$$

where the reflection angle is $\theta_r = \arccos(\mathbf{n}\cdot(-\omega_i))$ and the transmission angle is $\theta_t = \arccos(\sqrt{1-\eta^2(1-\cos^2\theta_r)})$. The reflection amplitude A_r follows the Fresnel equation:

$$A_r = \frac{Z_1\cos\theta_r - Z_2\cos\theta_t}{Z_1\cos\theta_t + Z_2\cos\theta_r}, \quad A_t = 1 - A_r. \tag{7}$$

To determine whether a ray reflects or transmits, we sample a random variable $y \sim \mathcal{U}(0,1)$ from a uniform distribution and reflect if $(y < A_r)$.

Contrary to existing works, we use a microfacet distribution to capture surface roughness, modeling both specular and diffuse scattering. Rather than treating each surface as uniformly planar, it is represented as many small facets, each with its own normal. We adopt the GGX microfacet distribution [26], widely used in physics-based rendering [19]. In this distribution, a roughness parameter

governs the transition between diffuse and specular behavior, varying from 0 for a perfectly diffuse surface to 1 for a perfectly specular surface.

Transducer Sampling To increase the chance of returning a ray to the transducer, we cast secondary rays from each interaction point toward a center point of a randomly chosen transducer element. If a secondary ray intersects a boundary, we evaluate the BSDF again and factor in the probability of continuing in the chosen direction. Each ray's contribution is then weighted by the transducer's receive directivity function,

$$f_d(\omega_i) = \begin{cases} 0, & \text{if } |\alpha| > \alpha_c, \\ \frac{\alpha_c - |\alpha|}{\alpha_c - \alpha_m}, & \text{if } \alpha_m < |\alpha| \leq \alpha_c, \\ 1, & \text{if } |\alpha| \leq \alpha_m, \end{cases} \tag{8}$$

where α is the angle between ω_i and the transducer normal $\mathbf{n}$. Angles within α_m see full reception, beyond α_c see none, and intermediate angles are linearly scaled. This can consequently be integrated into equation 3.

Phase Calculation and Signal Processing When a ray arrives at the transducer, its pressure is stored in the time-domain signal of the corresponding element. To incorporate phase, we convolve with a sinusoid windowed by a Gaussian function [8]:

$$s(t) = \sin(2\pi f_c t)\, \exp\big(-t^2/\sigma\big), \tag{9}$$

where f_c is the central frequency and σ sets the pulse width. After all rays are processed, standard beamforming (delay-and-sum), demodulation, and log compression yield the final B-mode image.

3 Implementation details

The implementation of the ray tracing is built on top of the Mitsuba 3 software [12], a physics-based renderer for forward and inverse light transport simulation of natural images. Mitsuba is written in C++ with Python bindings and offers flexibility through its modular framework. Additionally, the light transport equations are fully differentiable, enabling inverse transport simulations. The software can be built for various renderer variants that run on either CPU or GPU. For this work, we use the GPU variant, which leverages NVIDIA's OptiX rendering framework [20].

To adapt Mitsuba for ultrasound simulation, we derived custom classes in Python and C++ for the emitter, sensor, memory block, reconstruction filter, film, and integrator. Further details about the Mitsuba framework can be found in its documentation[5].

[5] https://mitsuba.readthedocs.io/en/stable/

For beamforming and signal processing, we utilize the Ultraspy library [9]. The ray tracing simulations are executed on a desktop PC equipped with an Intel i7-7200 processor (20 cores) and an NVIDIA RTX 4070 Ti GPU. The code for this framework can be found in this repository[6].

4 Experiments

We compare our proposed simulator to the open-source system by Mattausch *et al.* [18] (referred to as the baseline). Both simulators generate a B-mode image for a synthetic scene and a spine phantom, and these images are compared with real ultrasound data acquired using a Siemens Juniper Acuson (5C2 convex probe, 5 MHz center frequency, 128 elements, 70°opening angle) with the object of interest submerged in a water bath. To ensure consistency between the real and synthetic setups, we perform a CT scan of the spine phantom, convert the scan into a mesh using ImFusion[7], and create a virtual scene in Blender[8]. For the synthetic scene, the object of interest is 3D-printed. Small misalignments remain, limiting the comparison to qualitative observations.

Our simulator employs a 25-angle plane-wave scheme ([-30°,30°]), 5 MHz center frequency, 128 elements, 90 dB dynamic range, 50 MHz sampling, 5 cycles, and a 4 mm elevational beam width usual in conventional ultrasound systems [23]. Rays are traced up to 20 cm, with a maximum of 10 bounces per ray. Each element emits 100,000 rays, resulting in approximately 13 million rays emitted into the scene simultaneously, completing in about 1 s. Where possible, the baseline uses matching parameters (e.g., 5 MHz center frequency). We set 2° for the cutoff and beamwidth angles in the directivity function.

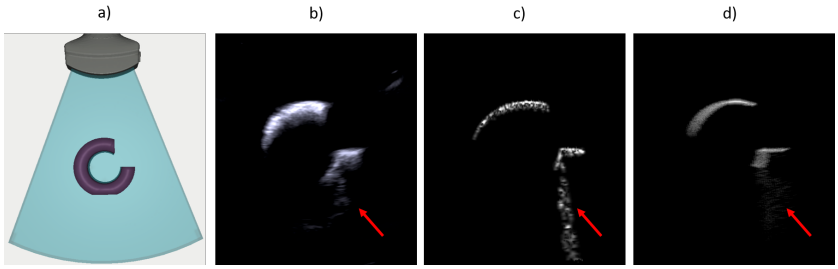

Fig. 3. Comparison of reflections created based on scanning a cylinder with a cutout as shown in a). b) shows a real acquisition, while c) represents the baseline and d) UltraRay.

[6] https://github.com/Felixduelmer/UltraRay
[7] https://www.imfusion.com/
[8] https://www.blender.org/

In the synthetic scene (Fig. 3), a cylinder with a cutout on one side was scanned, and its surface was configured to be perfectly diffuse. Due to the design of the cylinder, the likelihood of returning rays/waves to the transducer is small. This behavior is evident in both the real acquisition and our simulator, where only minor reflections appear beneath the opening. In contrast, the baseline model records an echo at every scene interaction, resulting in an unrealistically strong reflection below the opening.

For the vertebra, we apply the bone preset from the baseline [18], using an acoustic impedance value of 7.8 and set an acoustic impedance value of 1.54 for water (with zero scattering). Setting a roughness value of 0.5 in the microfacet distribution yields realistic bone reflections.

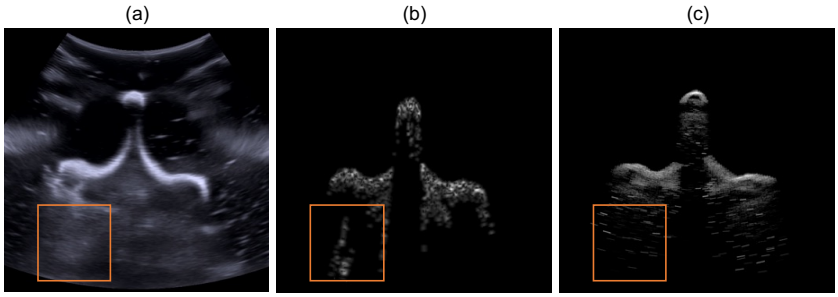

Fig. 4. Comparison of (a) real acquisition, (b) baseline simulator, and (c) UltraRay. The orange box highlights strong unrealistic reflections observed in the baseline, which are absent in both the real acquisition and UltraRay.

Figure 4 a) shows real data with noise and water-bath reverberations. Both simulators 4b) and c) capture the vertebra's geometry, although UltraRay more closely matches real intensities. The orange box highlights unrealistic reflections in the baseline simulator, produced when rays bounce off the surface above the highlighted region and intersect the spinous process (the elongated structure close to the transducer). Because this structure is nearly orthogonal to the transducer, those rays should exit the scene. However, the baseline simulator still records them as echoes, whereas they are absent in UltraRay and the real acquisition.

5 Discussion and Conclusion

The presented simulator lays a foundation for fast physics-based reflection modeling in ultrasound simulation using ray tracing. At its current stage, the simulator effectively models surface interactions at tissue boundaries and can be naturally extended to include scattering within heterogeneous tissue and attenuation, based on its ray tracing and transducer sampling framework. Currently,

we restrict the opening angle of transducer elements during the transducer sampling strategy to enhance image quality. While this constraint reduces noise, it also limits the simulator's ability to replicate realistic acoustic interactions, as real transducers with broader directivity can still capture off-axis returns and maintain visibility in partially occluded regions. Exploring methods to manage noise without sacrificing a wider opening angle would further improve the simulator's versatility and realism, making it more applicable to a broader range of ultrasound imaging scenarios.

In summary, we introduce UltraRay, a framework that integrates a Monte Carlo-based ray tracing method for enhanced reflection simulation in ultrasound. We propose a novel transducer sampling strategy that efficiently returns rays to the transducer after scene interactions. By incorporating a plane wave imaging acquisition scheme along with a conventional beamforming pipeline, we enable end-to-end ultrasound simulation. Built atop a modular, differentiable physics-based rendering framework, UltraRay establishes a foundation for future advancements in ultrasound simulation.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Amadou, A.A., Peralta, L., Dryburgh, P., Klein, P., Petkov, K., Housden, R.J., Singh, V., Liao, R., Kim, Y.H., Ghesu, F.C., et al.: Cardiac ultrasound simulation for autonomous ultrasound navigation. Frontiers in Cardiovascular Medicine **11**, 1384421 (2024)
2. Behboodi, B., Rivaz, H.: Ultrasound segmentation using u-net: learning from simulated data and testing on real data. In: 2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC). pp. 6628–6631. IEEE (2019)
3. Burger, B., Bettinghausen, S., Radle, M., Hesser, J.: Real-time gpu-based ultrasound simulation using deformable mesh models. IEEE transactions on medical imaging **32**(3), 609–618 (2012)
4. Burman, N., Manetti, C.A., Heymans, S.V., Ingram, M., Lumens, J., D'hooge, J.: Large-scale simulation of realistic cardiac ultrasound data with clinical appearance: methodology and open-access database. IEEE Access (2024)
5. Canney, M.S., Bailey, M.R., Crum, L.A., Khokhlova, V.A., Sapozhnikov, O.A.: Acoustic characterization of high intensity focused ultrasound fields: A combined measurement and modeling approach. The Journal of the Acoustical Society of America **124**(4), 2406–2420 (2008)
6. Clement, G., Hynynen, K.: Field characterization of therapeutic ultrasound phased arrays through forward and backward planar projection. The Journal of the Acoustical Society of America **108**(1), 441–446 (2000)
7. Dietrich, C.F., Lucius, C., Nielsen, M.B., Burmester, E., Westerway, S.C., Chu, C.Y., Condous, G., Cui, X.W., Dong, Y., Harrison, G., et al.: The ultrasound use of simulators, current view, and perspectives: Requirements and technical aspects (wfumb state of the art paper). Endoscopic Ultrasound **12**(1), 38–49 (2023)

8. Duelmer, F., Simson, W., Azampour, M.F., Wysocki, M., Karlas, A., Navab, N.: Phocus: Physics-based deconvolution for ultrasound resolution enhancement. In: International Workshop on Advances in Simplifying Medical Ultrasound. pp. 35–44. Springer (2024)
9. Ecarlat, P., Carcreff, E., Varray, F., Liebgott, H., Nicolas, B.: Get ready to Spy on Ultrasound: Meet ultraspy. In: International Ultrasonics Symposium (IUS). pp. 1–4. IEEE (2023)
10. Gao, H., Choi, H.F., Claus, P., Boonen, S., Jaecques, S., Van Lenthe, G.H., Van der Perre, G., Lauriks, W., D'hooge, J.: A fast convolution-based methodology to simulate 2-dd/3-d cardiac ultrasound images. IEEE transactions on ultrasonics, ferroelectrics, and frequency control **56**(2), 404–409 (2009)
11. Ghanem, M.A., Maxwell, A.D., Kreider, W., Cunitz, B.W., Khokhlova, V.A., Sapozhnikov, O.A., Bailey, M.R.: Field characterization and compensation of vibrational nonuniformity for a 256-element focused ultrasound phased array. IEEE transactions on ultrasonics, ferroelectrics, and frequency control **65**(9), 1618–1630 (2018)
12. Jakob, W., Speierer, S., Roussel, N., Nimier-David, M., Vicini, D., Zeltner, T., Nicolet, B., Crespo, M., Leroy, V., Zhang, Z.: Mitsuba 3 renderer (2022), https://mitsuba-renderer.org
13. Jensen, J.A.: Ultrasound imaging and its modeling. In: Imaging of complex media with acoustic and seismic waves, pp. 135–166. Springer (2002)
14. Kajiya, J.T.: The rendering equation. In: Proceedings of the 13th annual conference on Computer graphics and interactive techniques. pp. 143–150 (1986)
15. Li, T.M., Aittala, M., Durand, F., Lehtinen, J.: Differentiable monte carlo ray tracing through edge sampling. ACM Transactions on Graphics (TOG) **37**(6), 1–11 (2018)
16. Lobo, M.D., Miravent, S., de Almeida, R.P.P.: Emerging trends in ultrasound education and healthcare clinical applications: A rapid review. Emerging Technologies for Health Literacy and Medical Practice pp. 263–287 (2024)
17. Lucka, F., Pérez-Liva, M., Treeby, B.E., Cox, B.T.: High resolution 3d ultrasonic breast imaging by time-domain full waveform inversion. Inverse Problems **38**(2), 025008 (2021)
18. Mattausch, O., Makhinya, M., Goksel, O.: Realistic ultrasound simulation of complex surface models using interactive monte-carlo path tracing. In: Computer Graphics Forum. vol. 37, pp. 202–213. Wiley Online Library (2018)
19. Nimier-David, M., Vicini, D., Zeltner, T., Jakob, W.: Mitsuba 2: A retargetable forward and inverse renderer. ACM Transactions on Graphics (ToG) **38**(6), 1–17 (2019)
20. Parker, S.G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., et al.: Optix: a general purpose ray tracing engine. Acm transactions on graphics (tog) **29**(4), 1–13 (2010)
21. Peng, B., Huang, X., Wang, S., Jiang, J.: A real-time medical ultrasound simulator based on a generative adversarial network model. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 4629–4633. IEEE (2019)
22. Pharr, M., Jakob, W., Humphreys, G.: Physically based rendering: From theory to implementation. MIT Press (2023)
23. Scholten, H.J., Weijers, G., de Wild, M., Korsten, H.H., de Korte, C.L., Bouwman, R.A.: Differences in ultrasound elevational beam width (slice thickness) between popular handheld devices. WFUMB Ultrasound Open **1**(2), 100009 (2023)

24. Treeby, B.E., Cox, B.T.: k-wave: Matlab toolbox for the simulation and reconstruction of photoacoustic wave fields. Journal of biomedical optics **15**(2), 021314–021314 (2010)
25. Velikova, Y., Azampour, M.F., Simson, W., Gonzalez Duque, V., Navab, N.: Lotus: learning to optimize task-based us representations. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 435–445. Springer (2023)
26. Walter, B., Marschner, S.R., Li, H., Torrance, K.E.: Microfacet models for refraction through rough surfaces. Rendering techniques **2007**, 18th (2007)
27. Wysocki, M., Azampour, M.F., Eilers, C., Busam, B., Salehi, M., Navab, N.: Ultra-nerf: Neural radiance fields for ultrasound imaging. In: Medical Imaging with Deep Learning. pp. 382–401. PMLR (2024)
28. Zhang, C., Wu, L., Zheng, C., Gkioulekas, I., Ramamoorthi, R., Zhao, S.: A differential theory of radiative transfer. ACM Transactions on Graphics (TOG) **38**(6), 1–16 (2019)