Efficient 2D ultrasound simulation based on dart-throwing 3D scatterer sampling

François Gaits[1], Nicolas Mellado[1], and Adrian Basarab[2]

[1]Institut de Recherche en Informatique de Toulouse, UMR CNRS 5505, Université de Toulouse, France

[2]Université de Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne,
CNRS, Inserm, CREATIS UMR 5220, U1206, Villeurbanne, France

Abstract—**Ultrasound image simulation is a well-explored field with the main objective of generating realistic synthetic images, further used as ground truth (e.g. for training databases in machine learning), or for radiologists' training. Several ultrasound simulators are already available, most of them consisting in similar steps: (i) generate a collection of tissue mimicking individual scatterers with random spatial positions and random amplitudes, (ii) model the ultrasound probe and the emission and reception schemes, (iii) generate the RF signals resulting from the interaction between the scatterers and the propagating ultrasound waves. To ensure fully developed speckle, a few tens of scatterers by resolution cell are needed, demanding to handle high amounts of data (especially in 3D) and resulting into important computational time. The objective of this work is to explore new scatterer spatial distributions, with application to 2D slice simulation from 3D volumes. More precisely, lazy evaluation of pseudo-random schemes proves them to be highly computationally efficient compared to uniform random distribution commonly used. A statistical analysis confirms the visual impression of the results.**

Index Terms—**Ultrasound imaging, simulation, low discrepancy, 3D slice, scatterer distribution**

I. Introduction

Ultrasound imaging is used in number of medical applications due to its non-ionising, real-time and low-cost characteristics. Therefore, a rich literature exists in the field of ultrasound, ranging from innovative acquisition modes to image reconstruction, processing and analysis methods. In this context, generating synthetic ultrasound images, also known as image simulation, plays a key role in the development and the validation of new algorithms, allowing access to ground truth data. Moreover, ultrasound simulation is also widely used to provide data for training the practicians [1].

The main idea behind ultrasound image simulation is to generate the radiofrequency (RF) signals, further used to beamform the image, resulting from the interaction between simulated ultrasound waves and a tissue mimicking map. The shape of the synthetic ultrasound waves is related to the geometry of the simulated probe and the emission strategy. The tissue map generation is based on the assumption that tissues are composed of small reflectors, called scatterers. The scatterers are smaller than the wavelength, thus enabling the diffusion of the ultrasound waves, similar to what happens in real tissues [2]. Note that alternative solutions to scatterer maps exist, such as image or texture-based approaches [3].

The generation of scatterer maps is controlled by their number, spatial distribution (e.g. following a regular grid [4], or a random distribution [2]), and amplitudes. For the latter, a standard approach [5], [6] is to generate a random amplitude for each scatter, usually following a zero-mean Gaussian distribution with spatially varying standard deviation [7]. This way of generating the scatterers' amplitudes requires the use of a map representative of the tissues to be simulated, e.g., a medical image acquired using MRI [8] or CT [9]. The number of scatterers to be generated is a crucial parameter to ensure fully developed speckle. It has been reported in the literature [10] that several scatterers (of the order of tens) should be generated per resolution cell. However, in practice this number varies a lot depending on the approaches [1], [11], and there is no study of the impact of this parameter depending on the spatial sampling strategy.

Given the scatterers, the ultrasound images are computed by simulating their interaction with ultrasound waves emitted by a virtual probe. Usually, the simulators model the pressure field occurring at the scatterer locations, e.g., as in Field II [7], [12] and SIMUS [13]. Other methods exist, such as Monte Carlo path tracing [14], or by convolving an interpolated version of the scatterer map onto a regular grid with spatially invariant or variant point spread functions [15]–[17].

This work focuses on the simulation of *multiple 2D ultrasound slices* from a 3D volume populated with tissue-mimicking scatterers, allowing to generate 2D image sequences from a single volume, as performed experimentally in number of applications [1], [11]. In this setting, a large amount of scatterers is required in the 3D volume, in order to simulate ultrasound images with fully developed speckle independently of the slice orientation. This is challenging for large volumes, from the perspectives of their storage and efficient extraction for a given slice thickness. To mitigate these issues, this paper explores different pseudo-random strategies to spatially sample the scatterers. We propose to compute the positions of the scatterers using a constant sequence of random positions replicated in each cell of a regular 3D grid, and generated on the fly when computing the probe slice. The results reported show the computational efficiency of this distribution compared to existing approaches. In contrast to other approaches, we show that fully developed speckle is obtained for a relatively low number of scatterers per resolution cell.

II. Background on scatterer sampling strategies

A key aspect of ultrasound simulation is the generation of the scatterers map. The scatterers' location plays an important role in the simulation process: scatterers store local information about the simulated tissues. Several criteria have

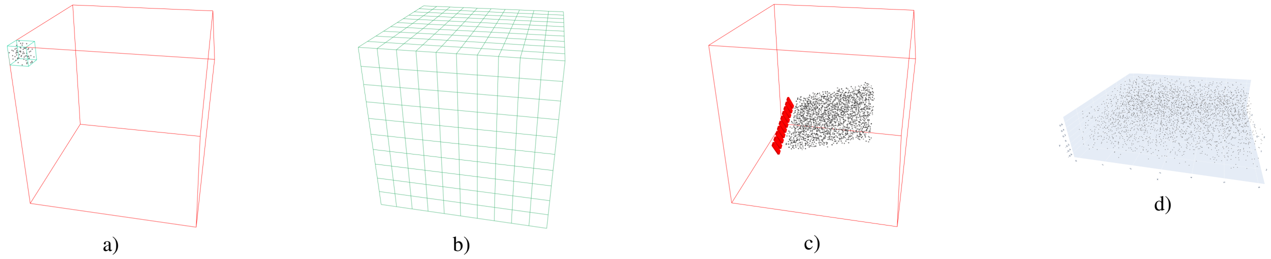


Fig. 1. Detail of the proposed framework: a) scatterers drawn for one cell, b) cells over the whole volume, c) generation and extraction of the scatterers corresponding to the probe position and characteristics, d) scatterers expressed in probe coordinates, to be used for 2D slice simulation.

been proposed to define the quality of scatterer maps: spatial density [14] (scatterers should not be too far from each others to avoid holes), alignment to density map [8], and shape of the simulated RF envelope [18].

Several strategies have been proposed to find optimal scatterers' locations. While the most common still remains the uniform random distribution [19], alternatives have been proposed in the literature to avoid the need for a large number of scatterers per resolution cell, to control the maximal distance between neighbouring scatterers or to improve their manipulation [2]. Among these strategies, one may cite the regular grids [4], repetitions of small distributions in large volumes [14], [20], or random location of one scatterer per small subvolume of the order of the resolution cell [21].

In this work, we study the analogy between scatterers' placement and point-based sampling in Monte Carlo integration. Monte Carlo integration is based on the idea of using random values to estimate the integral of a function: provided that enough values are available, this kind of estimator converges towards the true value of the integral. However, the quality of the random values greatly affects the convergence rate. Quasi-Monte Carlo methods use quasi-random or pseudo-random sequences that keep the advantages of a random estimator while providing properties that allow the estimation to converge faster, i.e., for less data available.

Analogously, we propose to study how the pseudo-random sequences used in Monte Carlo simulation improves the placement of scatterers in 2D or 3D grids. To this end, we propose to consider a specific criterion of the random sequence: the *discrepancy*. This measure quantifies the coverage of the domain by the chosen samples, i.e., the lower the discrepancy, the better the space coverage. As for Monte Carlo integration approaches, ultrasound simulation requires that the samples (or scatterers) location cover the entire domain without holes. This can be measured by comparing with how closely related they are with the uniform random distribution. Discrepancy can be expressed in several ways ; here, we consider the ℓ_2 star discrepancy, a standard metric for random sequences [22].

Interestingly, the discrepancy of a set of samples obtained with a given sampling strategy might evolve when the number of samples varies. Sequences (set of n points generated) which converge towards a discrepancy value of 0 will yield the exact result on integration, given enough samples. Moreover, sequences whose discrepancy decreases in $O(log(n)^{s-1}/n)$, with s the dimension and n the number of samples, are usually called *low discrepancy sequences* [23].

III. Proposed Method

The main contribution of this work is an approach to populate a 3D medium with scatterers, requiring low storage and enabling a computationally efficient extraction of the scatterers contributing to the simulation of 2D slices, as illustrated in Figure 1 and resumed in Algorithm 1.

As illustrated in 2D in Figure 2(a-c), different strategies may be employed to distribute the scatterers in a volume, leading to different types of artefacts: a) using a random distribution, which may cause holes, b) following a uniform grid [4], which requires a large number of scatterers, or c) by repeating the same random sequence in all cells, which generates patterns.

To mitigate these issues, we took inspiration from the theory of low discrepancy sequences [22], [24], [25]. The proposed strategy, illustrated in Figure 2(d), consists in repeating modified versions of a single random sequence. More precisely, starting from a single sequence representing 3D locations at the cell level, we populate all the cells using the same sequence but perturbed with random rotations. Hence, the cells do not store the sample location, but rather hold the random rotations: this reduces storage requirements and allows the generation of the scatters on the fly when slicing the volume. Furthermore, the grid of cells serves as an acceleration structure to extract the scatterers lying in a slice with a given orientation and thickness, speeding up the extraction process and facilitating the manipulation of large scale volumes.

We hereafter detail our choices about the random sequence (see Sec. III-A), the modulation of the sequence to efficiently sample the entire volume (see Sec. III-B), and the extraction of scatterers to sample a slice of the volume (see Sec. III-C).

A. Random sequence for scatterers' placement

The choice of the random sequence used to fill the volume is a crucial point, with high impact on the final distribution of the scatterers and on the simulated ultrasound images. As explained before, one could choose to draw the positions of the scatterers from a uniform distribution, or to simply arrange them on a regular grid. However, choosing more carefully a strategy could yield better results, as supported by knowledge on Monte Carlo integration and low discrepancy sequences [22], [23], [24]. These low-discrepancy sequences are shown hereafter to yield useful properties in ultrasound simulation, while retaining a random aspect. Among the existing approaches such as Halton [24] or Sobol [25], we study in this work a 3D generalization of Poisson disk sampling known as relaxed dart-throwing [26]. This approach follows a simple

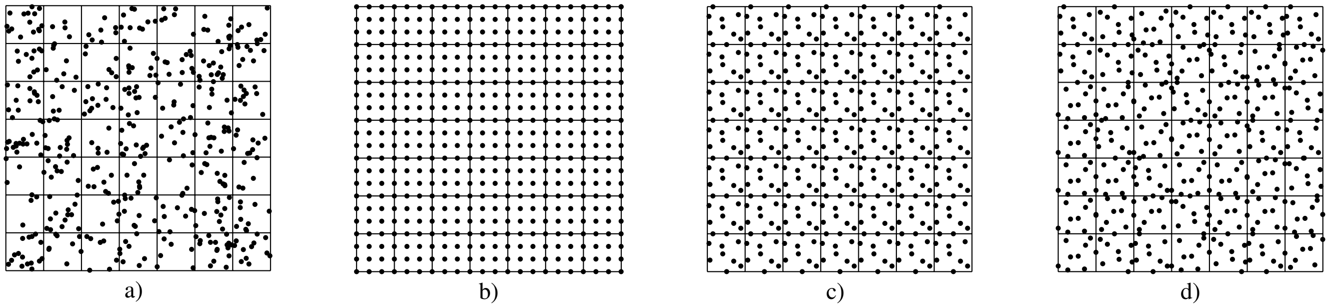


Fig. 2. Example of different strategies for distributing nine scatterers per mm^2 (441 samples in total): a) uniform distribution, b) regular grid, c) dart throwing without rotations, d) proposed dart throwing with rotations.

Algorithm 1: Slices extractions from a volume

```
Data:
$d :=$ Scatterer density
$v :=$ Volume information
$probeParams :=$ List of probe positions and orientation
Result:
$Slices :=$ Scatterers collections to be used for simulations
$G \leftarrow createGrid(v, d)$
foreach $p \in probeParams$ do
    $cells = G.getCellsCrossed(p)$
    $slice = empty()$
    foreach $cell \in cells$ do
        $points = cell.generatePoints()$
        foreach $point \in points$ do
            if $p.isInside(point)$ then
                $slice = slice + point$
    $Slices = Slices + slice$
```

heuristic, which rejects points that are too close to each others until the given number is attained. If too many points are rejected, the rejection radius is reduced.

B. Efficient volume sampling using sparse cell representation

Given a random sequence, our goal is to generate scatterers sampling the entire volume. A first naive approach would consist in generating and storing all the scatterers once for all. In 3D, this approach has a cubic space complexity with respect to the cell size.

We propose to use a *stratified sampling* strategy, where the same sequence is repeated in each cell of a grid covering the entire volume. Here, the sequence is computed once for all, and the scatterers are generated on the fly, only for the cells intersecting the simulation area, as described in the next section. In order to avoid sampling patterns, each cell is assigned with a rotation randomly generated among the 24 possibilities (rotations of the cube, modulo $\pi/2$), as illustrated in Figure 2-d). The amplitudes of the scatterers are obtained using a zero-mean random Gaussian generator with variance proportional to the local properties of the 3D medium to be simulated. In order to ensure consistent scatterer properties across consecutive runs, each cell is also associated with a constant seed used by the random number generator.

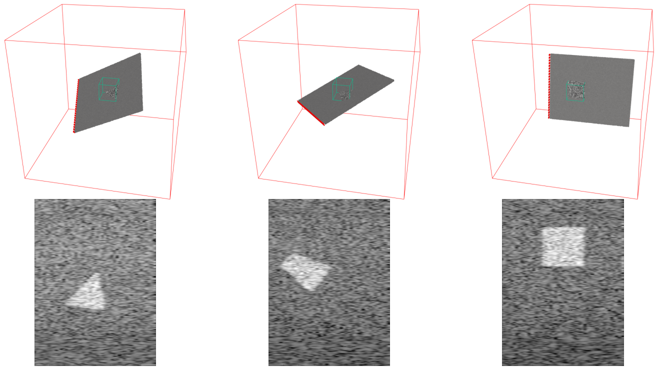


Fig. 3. 3D slices and resulting simulations using Field II.

C. Scatterer extraction by volume slicing

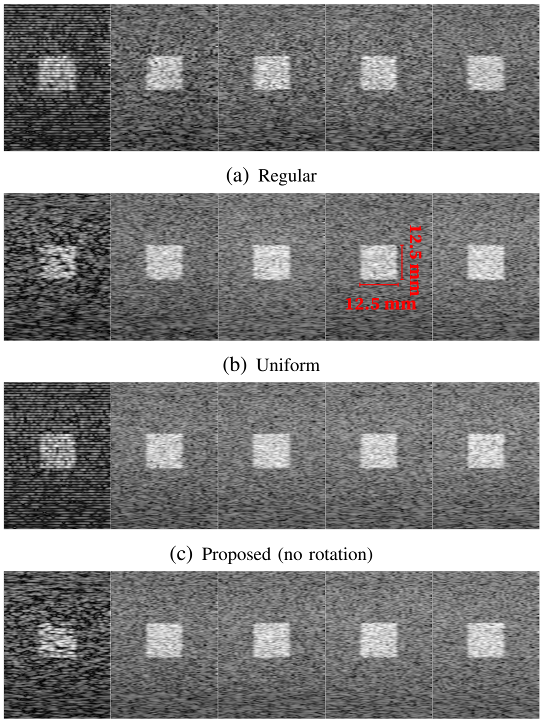
To simulate 2D ultrasound images (slices) from a volume populated with scatterers, an extraction of the scatterers within the probe field of view is necessary. We define the acquisition zone with respect to the position of the probe, and the thickness of the slice, which may vary depending on the type of probe to be simulated. To extract the scatterers, the proposed algorithm (detailed in Algorithm 1) first identifies the cells intersected by the acquisition zone. For each cell that is completely contained in the acquisition zone, i.e., its eight corners are inside, the scatterers are directly extracted without any further test. For the cells that are partially belonging to the field of view, each scatterer is tested before extraction, by projecting its spatial coordinates p in the probe coordinate system (denoted p_a), as follows:

$$p_a = (p - t_a) \times r_a^{-1},$$

with t_a, r_a the translation and rotation of the probe, respectively. In the probe coordinate system, the acquisition zone is an axis aligned box, against which the scatterers positions can be checked directly.

IV. Results

We implemented our algorithm using the UTK library [27], executed on AMD Ryzen 9 5900X 3.70Ghz (single thread), with 32go RAM. Simulation results were obtained using Field II [7], [12], by simulating a linear probe with 192 elements and center frequency of 3.5MHz. We show in Figure 3 an example of a 3D volume and three different ultrasound images



(a) Regular

(b) Uniform

(c) Proposed (no rotation)

(d) Proposed

Fig. 4. Comparison of simulation results obtained with Field II [12] for different distribution schemes, using 1, 8, 27, 64 and 125 scatterers per mm^3.

simulated from different view points. For all our experiments, we used 1 mm^3 cells, which is of the order of the axial resolution cell, thus allowing an arbitrary position of the probe relative to the simulated volume. In other words, the axial direction, i.e., the propagation of the ultrasound waves, can be simulated in any spatial direction while still ensuring fully developed speckle. We evaluate our approach considering visual (Section IV-A), statistical (Section IV-B) and computational (Section IV-C) characteristics, in comparison with the following baselines: scatterers distributed uniformly in the whole volume, scatterers distributed with stratified sampling (using dart-throwing sequences, but no rotation), and scatterers positioned on a regular grid.

A. Visual comparisons

In this experiment, we model a 3D volume as a cube in a homogenous medium, and extract a slice traversing the cube and aligned with the xy plane. We report in Figure 4 the simulation results for each approach for varying scatterer densities (from 1 to 125 scatterers per mm^3). We observe that dart-throwing produces stable and satisfying results even in low density distribution. In contrast, the other approaches produce specific artefacts. For instance, when using 8 scatterers per mm^3 (second row), regular sampling leads to visible patterns, while uniform sampling produces dark holes due to irregular local space coverage.

B. Statistical analysis

In this section, we compare the statistical properties of the simulated images. Taking inspiration from [18], we analyze the RF envelope of the simulated images by measuring their divergence reported to a target Rayleigh distribution. To this mean, we populate scatterers in a homogeneous 3D medium and extract an axis-aligned slice for simulation.

Algorithm 2: Estimation of target Rayleigh distribution D_r

```
$t \leftarrow 32$;
$n \leftarrow 343$;
$D \leftarrow$emptyDistribution();
**foreach** $i \in [0:t]$ **do**
    $S$ = getScatterers($n$);
    $I$ = simulate($S$);
    $D_i$ = computeDistribution($I$);
    $D = D + D_i$;
$D_r$ = RayleighFit( $D/t$ );
```

Density	Regular	Uniform	Proposed (no rotation)	Proposed
1	0.12	0.05	0.12	0.12
8	0.12	0.5	0.12	0.12
27	0.13	1.69	0.13	0.13
64	0.14	3.91	0.14	0.14
125	0.16	7.70	0.16	0.16
343	0.28	22.89	0.28	0.29
512	0.36	31.75	0.37	0.38

TABLE I
EXTRACTION TIME (SECONDS) RELATIVE TO SCATTERER DENSITY IN A 100X100X100 MM^3 VOLUME

According to the visual comparisons shown in Figure 4, we assume that each sampling strategy converges to a reference solution when increasing the number of samples. Hence, we analyze the ultrasound images when increasing the number of scatterers, by measuring the Kullback–Leibler divergence between each realization and a target Rayleigh distribution D_r. As each strategy might lead to different results, we propose to compute a different target Rayleigh distribution for each sampling strategy. As both the scatterers sampling and the simulation processes are non-deterministic, we compute each target distribution by combining multiple simulation results, as described in Algorithm 2. We report in Figure 5-left the evolution of the KL divergence for each approach, with mean and variance values for 32 realizations. While the KL divergence decreases for all approaches, our approach converges faster than the three others, especially for small numbers of samples (< 8), where the divergence is overall one order of magnitude smaller than the regular grid. We also report in Figure 5-right the ℓ_2 discrepancy of each approach: our method has a better average discrepancy than the regular grid or uniform sampling strategies.

C. Performance analysis

We showed in the previous sections that our approach produces better but comparable results than the uniform distribution, and that these two approaches outperform the regular grid. As shown in Tables I and II, the uniform approach is

Thickness	Regular	Uniform	Proposed (no rotation)	Proposed
1	0.12	3.91	0.12	0.13
2	0.14	3.91	0.13	0.14
3	0.15	3.94	0.15	0.16
4	0.17	3.94	0.16	0.17
5	0.18	3.93	0.17	0.19

TABLE II
EXTRACTION TIME (SECONDS) RELATIVE TO SLICE THICKNESS (MM) FOR A FIXED (64) DENSITY

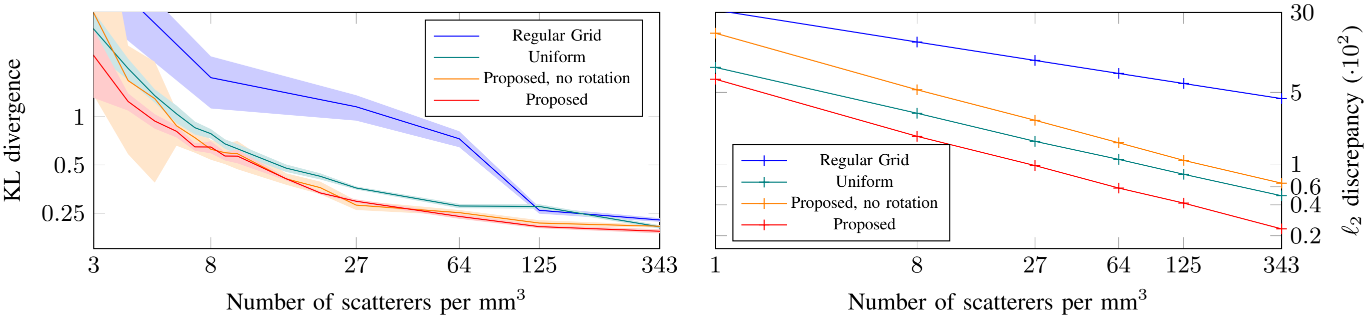


Fig. 5. Evolution of KL divergence (left: mean and variance, 32 runs) and ℓ_2 star discrepancy (right: mean, 100 runs) wrt. the number of scatterers (logscale).

several order of magnitude slower than our approach and the regular grid, for volume sampling and slice extraction. In terms of memory requirement, our approach is a stratified sampling strategy that does not require storing the scatterers in the grid, and thus largely outperforms the other approaches.

V. Conclusion

In this paper, we demonstrate the benefit of using stratified sampling strategies for the generation and extraction of scatterers for ultrasound image simulation. We specifically target the generation of multiple slices from a volume, and show how it outperforms existing approaches in terms of visual and statistical properties of the simulated images, as well as memory and computational requirements.

References

[1] Sjur Urdson Gjerald, Reidar Brekken, Torbjorn Hergum, and Jan D'hooge, "Real-time ultrasound simulation using the gpu," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 59, no. 5, pp. 885–892, 2012.

[2] John W. Hunt, Arthur E. Worthington, and A. T. Kerr, "The subtleties of ultrasound images of an ensemble of cells: simulation from regular and more random distributions of scatterers.," *Ultrasound in medicine & biology*, vol. 21 3, pp. 329–41, 1995.

[3] Orcun Goksel and Septimiu E. Salcudean, "B-mode ultrasound image simulation in deformable 3-d medium," *IEEE Trans. on Medical Imaging*, vol. 28, no. 11, pp. 1657–1669, 2009.

[4] Ricardo Dantas, Eduardo Costa, and Sidney Leeman, "Ultrasound speckle and equivalent scatterers," *Ultrasonics*, vol. 43, pp. 405–20, 06 2005.

[5] Martino Alessandrini, Bidisha Chakraborty, Brecht Heyde, Olivier Bernard, Mathieu De Craene, Maxime Sermesant, and Jan D'Hooge, "Realistic vendor-specific synthetic ultrasound data for quality assurance of 2-d speckle tracking echocardiography: Simulation pipeline and open access database," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 65, no. 3, pp. 411–422, 2018.

[6] Oliver Mattausch and Orcun Goksel, "Image-based reconstruction of tissue scatterers using beam steering for ultrasound simulation," *IEEE Trans. on Medical Imaging*, vol. 37, no. 3, pp. 767–780, 2018.

[7] Jørgen Arendt Jensen, "Field: A program for simulating ultrasound systems," in *10th Nordic-Baltic Conference on Biomedical Imaging Published in Medical & Biological Engineering & Computing, Volume 34, Supplement 1, Part 1*, 1996, pp. 351–353.

[8] Renaud Morin, Björn Eiben, Luc Bidaut, John Hipwell, Andrew Evans, and David J. Hawkes, "3d ultrasound simulation based on a biomechanical model of prone mri in breast cancer imaging," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015, pp. 264–267.

[9] Jean-Louis Dillenseger, Soizic Laguitton, and Eric Delabrousse, "Fast simulation of ultrasound images from a ct volume," *Computers in biology and medicine*, vol. 39, pp. 180–6, 02 2009.

[10] N. Rao, S. Mehra, and H. Zhu, "Ultrasound speckle statistics variations with imaging systems impulse response," in *IEEE Symposium on Ultrasonics*, 1990, pp. 1435–1440 vol.3.

[11] Adrien Marion and Didier Vray, "Toward a real-time simulation of ultrasound image sequences based on a 3-d set of moving scatterers," *IEEE Trans. on ultrasonics, ferroelectrics, and frequency control*, vol. 56, pp. 2167–79, 10 2009.

[12] Jørgen Jensen and Niels Svendsen, "Svendsen, n.b.: Calculation of pressure fields from arbitratily shaped, apodized, and excited ultrasound transducers. ieee trans. ultrason. ferroelectr. freq. control 39(2), 262-267," *IEEE Trans. on ultrasonics, ferroelectrics, and frequency control*, vol. 39, pp. 262–7, 02 1992.

[13] Amanda Cigier, François Varray, and Damien Garcia, "Simus: an open-source simulator for ultrasound imaging. part ii: comparison with three popular simulators," 2021.

[14] Oliver Mattausch and Orcun Goksel, "Monte-Carlo Ray-Tracing for Realistic Interactive Ultrasound Simulation," in *Eurographics Workshop on Visual Computing for Biology and Medicine*, Stefan Bruckner, Bernhard Preim, Anna Vilanova, Helwig Hauser, Anja Hennemuth, and Arvid Lundervold, Eds. 2016, pp. 173–181, The Eurographics Association.

[15] H. Gao, H.F. Choi, P. Claus, S. Boonen, S. Jaecques, G. Harry Van Lenthe, G. Van Der Perre, W. Lauriks, and J. D'hooge, "A fast convolution-based methodology to simulate 2-dd/3-d cardiac ultrasound images," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 2, pp. 404–409, 2009.

[16] Torbjørn Hergum, Stian Langeland, Espen W. Remme, and Hans Torp, "Fast ultrasound imaging simulation in k-space," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 6, pp. 1159–1167, 2009.

[17] F. Varray, O. Basset, Piero Tortoli, and Christian Cachard, "CREANUIS: A Nonlinear Radio Frequency Ultrasound Image Simulator," *Ultrasound in Medicine & Biology*, vol. 39, no. 10, pp. 1915–1924, 2013.

[18] Lin Zhang, Valery Vishnevskiy, and Orcun Goksel, "Deep network for scatterer distribution estimation for ultrasound image simulation," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. PP, pp. 1–1, 08 2020.

[19] J.A. Jensen and I. Nikolov, "Fast simulation of ultrasound images," in *2000 IEEE Ultrasonics Symposium. Proceedings. An International Symposium (Cat. No.00CH37121)*, 2000, vol. 2, pp. 1721–1724 vol.2.

[20] Benny Burger, Sascha Bettinghausen, Matthias Radle, and Jürgen Hesser, "Real-time gpu-based ultrasound simulation using deformable mesh models," *IEEE Trans. on Medical Imaging*, vol. 32, no. 3, pp. 609–618, 2013.

[21] Sjur Gjerald, Reidar Brekken, Torbjørn Hergum, and Jan D'hooge, "Real-time ultrasound simulation using the gpu," *IEEE Trans. on ultrasonics, ferroelectrics, and frequency control*, vol. 59, pp. 885–92, 05 2012.

[22] Hélène Perrier, *Anti-Aliased Low Discrepancy Samplers for Monte Carlo Estimators in Physically Based Rendering*, Ph.D. thesis, 03 2018.

[23] Harald Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, Society for Industrial and Applied Mathematics, USA, 1992.

[24] J.H. Halton, "On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals.," *Numerische Mathematik*, vol. 2, pp. 84–90, 1960.

[25] I.M Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 4, pp. 86–112, 1967.

[26] Michael Mccool and Eugene Fiume, "Hierarchical poisson disk sampling distributions," *Proceedings - Graphics Interface*, 05 1992.

[27] "Uni(corn—form) tool kit," https://utk-team.github.io/utk/.