PROCEEDINGS of the
23rd International Congress on Acoustics
9 to 13 September 2019 in Aachen, Germany

An attention-guided algorithm for improving the performance of acoustic simulations

Hanna AUTIO[(1)], Delphine BARD-HAGBERG[(2)]

[(1)]Lund University, Sweden, hanna.autio@construction.lth.se
[(2)]Lund University, Sweden

Abstract

When performing acoustic simulations with the purpose of auralization, there is a trade-off between accuracy and speed. In real-time simulations of virtual reality, finding the balance of this trade-off is paramount to achieving the desired result. If successful, the simulation speed is sufficient to provide a seamless acoustic experience as the agent moves around the space, while still being accurate enough to be realistic. It is generally accepted that a 20ms update interval for the impulse response is sufficient for achieving proper interactivity in most applications. However, reaching this threshold without degrading the quality of simulation too badly can be challenging indeed for complex scenes.

In this paper, a compromise between interactivity (response time) and accuracy is suggested for raytracing simulations. This compromise mimics the behaviour of the listener or the agent, prioritizing speed or accuracy based on how the agent behaves. When the listener is actively moving around the space, interactivity is prioritized. When the listener stands still, fully immersing in the experience, accuracy is improved. This is achieved by exploiting a fundamental truth of Monte Carlo simulations: Convergence improves with more samples.

Keywords: Raytracing, Real-Time, Auralization

1 INTRODUCTION

The VR and AR markets are continuing to grow, and both hardware and computation algorithms are improving quickly. Sound adapted for headtracking is becoming more and more relevant, and the framework for Ambisonics encoding has been on Youtube [1] for years. Despite all these advances, reliable and accurate programs for simultaneous real-time sound spatialization and localization are scarce, especially when compared to the corresponding state of the art for visual representations.

There are a number of ways that sound and video can be generated digitally, ranging from offline and static to real-time and interactive. Using an offline simulation strategy is a good method to achieve impressive quality and level of detail, and some level of offline simulation or rendering is used in almost all applications for both auditory and visual aspects. Traditional animated movies are examples of the extreme case, where the full experience is generated in advance. The issue with doing this is the lack of interactivity. If everything is generated in advance, in the absence of an agent, there is no way to allow the agent to change the outcome of the simulation. In the other extreme, with full realtime algorithms, the interactivity is maximal, and any actions made by an agent can affect the results of the simulation. For interactive medias such as video games, some middle ground between these extremes is usually the method of choice.

While there are similarities in the principles of how audio and graphics are generated, there are large differences in practice. Often, the level of precalculation is much higher for the sound field, and the opportunities for interaction with the sound field at large is quite limited. Indeed, there are few ways of generating plausible and physically based auralizations in realtime, and most of the commonly used programs focus on only one of spatialization or localization, rather than the complex interplay between these. As a consequence, most sound simulations are generated primarily offline, with a fairly low level of interactivity. In order to fully exploit the advantages of VR and AR, faster and better methods should be developed.

The discrepancy between how the modalities of vision and hearing are treated in interactive media is mirrored

in how developed the algorithms and hardware for the corresponding simulations are. Dedicated hardware designed specifically for the efficient calculations of graphics simulations are commonplace. However, this offers advantages also to acoustic modelling strategies, as long as they can exploit the same algorithms used for graphics simulations. One of these algorithms is raytracing, which can be used both for audio and video simulations and calculations.

This paper is part of an effort in developing a realtime raytracing engine, which can be used to create auralizations of complex environments for VR. One of the issues with raytracing is that it is difficult to generate satisfactory room impulse responses quickly enough for the simulations to be realtime. This paper presents a method that could potentially be used to mediate the effects of compromise between response rate and quality. The method is still developing, and some of the foreseen issues in its implementation are presented.

The idea is based on attempting to infer the attention of human agents in an interactive environment, so the first section will briefly discuss human attention. Subsequently, some background regarding Monte Carlo simulations in general and raytracing in particular is given. On this foundation, the idea is presented, and some of the foreseen issues are discussed.

2 AUDITORY ATTENTION

In this paper, it is supposed that agents in an environment will direct their attention to different focal points at different times. This implies that the agents attend to different discrete objects at different times, but also that agents' attention will shift towards or away from different phenomenons or modalities. For example, it is assumed that an agent who pays close attention to the task of navigating the space will have less of their attention focused on the acoustic perception of the room (provided they are of normal sensibilities). Understanding and exploiting the mechanisms of attention can be key to developing more efficient simulation algorithms.

The concept of auditory attention is not new, nor debated. Phenomena such as the cocktail party effect clearly show that humans are capable of directing their auditory attention to a specific point in space, and there is research showing that attention can be guided towards specific sound patterns as well [2]. Traditionally, assumptions regarding the focus of attention are made in order to determine what objects or sounds should be more carefully and accurately modelled.

What is novel in the approach described here is that, rather than inferring what sounds or auditory events are most important, the aim is to determine when sound in general is more or less important for the overall experience. This might be possible if there are variations in how much an agent attends to different senses. The theory of limited attentional bandwidth states that there is a limit to the amount of things humans can attend to at once. While the concept is debated, evidence supporting it can be found in research regarding things such as texting while walking. While the limited bandwidth phenomenon seems to be most influential when the tasks interfer with the same modality, or sense, there is research indicating that deviations from what is expected in a non-attended modality are more likely to go undetected [4].

If it is true that full attention can not be placed on all aspects of an experience at once, lack of attention for one aspect can be inferred by increased attention or focus on something else. As an example, if an agent closes their eyes and stands still, it may indicate an increased attention to the auditory experience. Conversely, an agent walking briskly towards a specific visual element is likely not attending whole-heartedly to the sound field. The assumption that this type of interplay exist is the basis for the algorithm suggested in section 4.

3 SOUND SIMULATION WITH RAYTRACING

Ray tracing is a widely used method of sound field simulation. It is theoretically founded on a model stating that the sound field generated by a sound source can be accurately described using an infinite number of sound particles, each carrying a finite amount of energy. The sound particles are emitted from the source, according to its directional characteristics, and when unobstructed travel along straight paths until they are annihilated. As a sound particle encounters an obstacle, it is reflected in a direction determined by a physically motivated

distribution. At any given time, the sound field generated by the source can be evaluated by calculating the energy in the sound field, given by the all sound particles.

This theoretical model becomes useful by virtue of the Monte Carlo framework. It gives a way to estimate the transfer function between two points by only studying a subset of the infinitely many sound particles. In the following sections, some comments will be made regarding Monte Carlo simulations.

To avoid ambiguities, in the following sections "source" will refer to a sound source, "listener" to a sound listener, "emitter" to a simulation object which emits rays and "receiver" a simulation object which can be hit by rays.

3.1 Some background on Monte Carlo simulations

Monte Carlo simulations are frequently used in many research areas, and are a way to estimate some unknown stochastic value. This is performed by drawing many samples of random variables which are somehow related to what should be estimated. In the context of acoustic raytracing for auralization, the goal of the algorithm is to derive an estimate of the room impulse response based on repeated sampling of ray paths between a sound source and a sound listener.

The Monte Carlo framework is entirely based on being able to sample the random variables that make up the end result. The results of the simulation depends entirely on the quality of the calculations used to produce the random samples, and much of the research in the area is aimed at developing improved sampling strategies. For acoustic raytracing, the sampling method is the generation of the path of each ray. There are many variations, some more random and some less. What model is best depends on the sound field and what questions the simulation aims to answer.

Samples are drawn according to the model distributions until the final estimate can be said to have converged. If the Monte Carlo simulation is well-designed, it will always converge to the true estimate. Studying how it converges is the foundation for most methods of error estimation.

3.1.1 Error estimation

As Monte Carlo simulations are a way of estimating the true value of a stochastic variable, it is fundamentally impossible to know whether the estimate is correct. With this in mind, the method would be useless without ways of approximately determining the error in the estimate, or without a certain amount of predictability.

One of the most basic, but also quite reliable, ways of knowing if a Monte Carlo simulation is good enough is to observe how much it varies between runs. Running several iterations of the same simulation and achieving the same result can then be considered an indication that it has converged. This can be used to estimate how many samples may be necessary, by running several simulations with a large number of samples and tracking the estimate over time. The minimum number of samples for which the estimate seems to have converged for each simulation can be an indication of the number of samples needed for convergence.

This requires that the samples are somehow random. Using a raytracer with a deterministic ray distribution and no random elements in the scattering will always yield the exact same result after a given amount of samples. This is true also if the same random seed is used in each iteration.

3.2 Implementation strategies for the raytracer in this project

The development of a raytracing simulation framework requires a number of choices to be made, both regarding physical model and practical implementation. This paper is written as a part of a larger project, and some decisions that have been made for the raytracer in this project are discussed here. In particular, the decisions that should affect the efficiency of the suggested model are discussed.

The raytracer in question implements an algorithm for *diffuse rain*, meaning that the number of rays can be reduced significantly compared to traditional raytracing [3]. The diffuse rain model uses some principles from radiosity theory to modify the way ray paths are sampled. Any time a ray undergoes a scattering reflection, its energy is dispersed according to some function. Some of this energy is then reflected towards each receiver in

the space, and the energy that hits a receiver this way is added to the calculated room impulse response. Doing this should ensure that each emitted ray contributes to the impulse response, and it should increase probability that important early reflections are included in the calculation. From a Monte Carlo-perspective, the implications are more complicated. The secondary energy transfer from reflection point to receiver can be considered as a sample of an additional ray, but this ray is not statistically independent from all other samples. This will change the distribution of samples in a complicated way, and while it should not introduce any incorrect samples it may affect the result. In performing the calculations for diffuse rain, it is also important to ensure that the energy level is preserved throughout the trace.

The room impulse response generated using these simulations are for a given listener-source position combination. These are symmetric, meaning that transmission from a given source position to one listener position is equivalent to transmission in the opposite direction, or that the simulation is insensitive to which of these two positions is used as emitter or receiver. In this particular project, the goal of the simulation is to give a single listener an experience with several sound sources. For this reason, the raytracing algorithm uses the listener as ray emitter. Together with the diffuse rain algorithm, this means that a single trace run will give the transfer function for all the different source positions to the single listener. If there were instead multiple listeners and few sources, the opposite choice might have been better.

Each ray transmission path yields one sample for the full room impulse response, and the distribution of samples is governed by many complicated principles. One of the factors which most clearly affect the distribution of samples is the choice of how the rays are generated, and in which direction they are initially emitted. One choice is to generate them in a deterministic manor, with the location and direction of each ray determined in advance. An example would be to generate rays over an equidistant grid on a sphere. If an infinite number of rays were generated, this would be a good way to approximate an omnidirectional source or listener. In the more general case, when less rays are used, using a deterministic grid such as the one described above could introduce systematic errors and may influence error estimation. Instead this project uses an algorithm which emits rays in a random direction. In the case of an omnidirectional listener, a uniform distribution over a sphere is used. The distribution of rays can be modified as is needed, to better conform to the requirements of the simulation and to better match the listener.

When a ray is generated, its direction and its energy should be determined. In this project, each ray has equal energy. If the object represented by the emitter has some energy directivity pattern, this should be implemented by varying the probability density pattern of ray generation so that more rays are emitted in this direction. Doing it this way ensures that each sample is as important for the aggregate simulation response, rather than letting some path samples be more important without necessarily being more likely. In general, there are many options for how to generate the ray directions and it is important to consider how the choices may influence especially the directional patterns of the generated impulse responses.

4 THE SUGGESTED ALGORITHM AND ITS IMPLEMENTATION

After the presentation above, it is time to connect the statements made regarding human attention and the raytracing algorithm. The interplay between these two aspects of real-time auralization can (and should) be used to develop better simulation strategies.

The highest demands on update frequency are made when the listening agent is rapidly moving about the space, as this corresponds to the times when the sound field actually changes between frames. On the other hand, based on the discussion in section 2, this should correspond to times when the demands on quality are low, as the agent focuses on interaction. Conversely, the highest demands on auralization quality should coincide with the times when the demands on update frequency are lowest, for agents moving naturally through the virtual space.

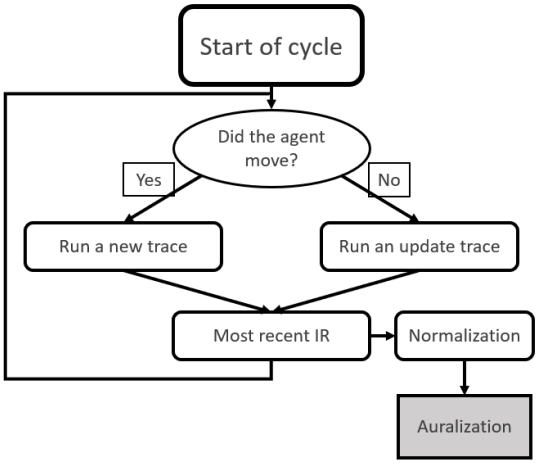

Figure 1 – Flowchart illustrating the method of raytracing presented in this paper.

4.1 The suggested framework

The suggested solution is this: If the agent has not moved significantly since the last frame, the impulse response used in the previous time step should be updated with the results from a new trace. If the agent has moved, the impulse response should instead be replaced with the results of a new simulation. A flowchart showing the algorithm is shown in fig. 1.

This would allow for a smaller number of rays in each time frame. When the agent is not moving, the results of the simulation is improved over time as more samples are added, which will lead to a high-quality and immersive experience. The update traces are always only intended to improve the results of an already acceptable impulse response, so there is no hard requirement on how many rays are needed. Any number of rays leads, on average, to an improved simulation result. In the case of an initial simulation, that is not an update trace, the requirements must be higher. The results of these traces must be sufficiently accurate to be plausible and predictable. However, these requirements can be significantly lower if it is not expected that the listener will be wholeheartedly listening to the sound field.

In conclusion, each iteration of the raytracing algorithm can use fewer rays so as to increase the update frequency, while the simulation as a whole still reaches the optimal quality when needed.

The algorithm as suggested fits well into the Monte Carlo simulation framework, and the basic principles of simulation stays the same. What it may change, however, is how the demands on quality is translated into simulation decisions and limitations. For example, it introduces the need for an explicit spatial discretization.

4.2 Spatial and temporal resolution

As much as possible, decisions regarding the simulation algorithm and strategy should be based on what the required performance is. In this case, this quite clearly includes the spatial and temporal resolution. Since these are intimately connected by sound propagation speed and the movement speed of the agent, these are discussed together.

Using the updated framework presented in fig. 1, the update frequency can remain stable and quite short. A temporal resolution of about 20 ms is usually considered realtime in acoustic contexts, and is consequently a natural suggestion as a time parameter. The issue with this time is that it is often too short to produce acceptable results from a raytracer (or other simulation software). As this algorithm is expected to have a lower demand on what is acceptable, it is possible that the number of rays needed is decreased to where 20 ms is sufficient. This is not clear at the moment, and in the meantime the other aspects which can affect the needed response time should also be considered.

If the algorithm should match the behavior of the agent, the resolution constants should also be chosen with the agent in mind. Human walking speed is approximately $1.4\,\mathrm{m\,s^{-1}}$. Depending on what spatial resolution is

needed, which will vary depending on the qualities of the sound field, this number can be used to determine what temporal resolution is needed. If spatial variations over a distance corresponding to the wavelength for 4 kHz (a common number in room acoustics) should be faithfully represented, the room impulse response should be updated after about 4.25 cm has been traversed. This corresponds to a time interval of about 30 ms. Of course, if the sound field is more diffuse, a more sparse resolution may be acceptable. When determining these factors, simulation results and estimates of the just noticeable differences should be used.

The suggested algorithm introduces an explicit spatial resolution in a way that is not traditionally the case for realtime simulations. This resolution is introduced to determine when an agent remains still and an update trace rather than a new trace should be performed. In general, it can not be expected that the agent is in exactly the same space between frames, as heartbeats, breathing, momentary loss of balance or health issues are likely to lead to small variations. These types of movements should be allowed within the spatial resolution. In some sense, this will lead to a spatial discretization, much in the same way as the temporal resolution leads to a time discretization.

There are two intuitively simple ways of introducing the spatial discretization. One is to precompute a grid with points in the model that are available to act as emitters or receivers. This method is quite close to what is often done for offline simulations, where a grid of transfer functions is computed in advance. In those cases, interpolation schemes are used to estimate the sound field between grid points. Something similar could be done in the current case, by updating the impulse response from one location by an update trace from an adjacent locations. The updated response is then a linear combination of the two, similar to what is used in many interpolation schemes. An issue with the grid-based discretization is that it requires significant computation beforehand.

The other option is to not generate a grid in advance, but simply replace the impulse response with a new one if the agent is far enough from the center location of the impulse response. This has the advantage of being fairly easy to implement, and easily applied to new models. The temporal resolution might need to be finer in this case, however, as it does not lend it self to interpolation methods as clearly.

The interpretation of updated responses as a linear combination of the previous response and the update trace is useful also in determining proper methods of energy normalization.

4.3 Energy normalization

In classical ray tracing simulations, rays of a total energy of 1 unit is emitted, corresponding to the total energy in an ideal impulse. If new rays are emitted as part of an update trace, the total energy level will no longer be equal to 1, and the response must be normalized. Due to the linearity of the system, this can be done by simply dividing the response by the total energy emitted. Below, two examples of normalization procedure is presented.

Assume that a first trace of n_1 rays is performed. Each ray has the same energy e_0 and the total energy $E_1 = n_1 \cdot e_0 = 1$. Further, let r_i denote the total energy that reaches the receiver from ray i. Then, the energy at the receiver after trace 1, denoted R_1, is

$$\sum_{k=1}^{n_1} r_k = R_1. \tag{1}$$

This does not need to be normalized, as $E_1 = 1$. If the impulse response is updated to be the result of n_2 rays with equal energy e_0, the new total emitted energy E_2 and received energy R_2 is

$$E_2 = n_1 \cdot e_0 + (n_2 - n_1) \cdot e_0 = n_2 \cdot e_0 \neq 1, \tag{2}$$

$$R_2 = R_1 + \sum_{k=n_1+1}^{n_2} r_k = \sum_{k=1}^{n_2} r_k. \tag{3}$$

In order to normalize this response to the total emitted energy, the proper normalization factor should be E_2. This gives the normalized energy contribution from ray i

$$\hat{r}_{i,2} = \frac{r_i}{E_2}. \tag{4}$$

This result can be used for auralization but the non-normalized values should be retained. The reason for why is shown below.

Suppose that the agent has not moved before the next frame, and a third trace should be added. The total number of emitted rays is n_3, and the total emitted energy is $E_3 = n_3 \cdot e_0$. With the same notation as before, the energy received before and after proper normalization is

$$R_3 = \sum_{k=1}^{n_3} r_k, \tag{5}$$

$$\hat{R}_3 = \frac{R_3}{E_3} = \sum_{k=1}^{n_3} \frac{r_k}{E_3} \tag{6}$$

In this case, there are no issues. If instead the results of trace 3 are added to the *normalized* results from trace 2, the situation will be different. $R_{3,N}$ denotes the total energy at the receiver, after adding the rays from trace 3 to the normalized value. Then

$$R_{3,N} = \hat{R}_2 + \sum_{k=n_2+1}^{n_3} r_k = \sum_{k=1}^{n_2} \frac{r_k}{E_2} + \sum_{i=n_2+1}^{n_3} r_i, \tag{7}$$

with total emitted energy

$$E_{3,N} = 1 + (n_3 - n_2)e_0. \tag{8}$$

Using this as a normalization factor will give a total emitted energy of 1. This gives the expression

$$\hat{R}_{3,N} = \frac{R_{3,N}}{E_{3,N}} = \tag{9}$$

$$\frac{1}{1 + (n_3 - n_2)e_0} \cdot \sum_{k=1}^{n_2} \hat{r}_k + \frac{1}{1 + (n_3 - n_2)e_0} \cdot \sum_{k=n_2+1}^{n_3} r_k = \tag{10}$$

$$\frac{1}{1 + (n_3 - n_2)e_0 \cdot E_2} \cdot \sum_{k=1}^{n_2} r_k + \frac{1}{1 + (n_3 - n_2)e_0} \cdot \sum_{k=n_2+1}^{n_3} r_k \tag{11}$$

Studying this expression clearly shows that different normalization factors are used for rays from different traces. Since the energy level increases with each subsequent trace, this will lead to a weighting where the most recent rays influences the impulse response more. This may or may not be the intended behaviour. If a grid is implemented as discussed in the previous section, this could be used to produce an impulse response which is a linear combination of the responses at two different grid points. Doing so should be done with great care, and the energy levels must be chosen in a way that provides reasonable coefficients from an interpolation perspective.

4.4 Ray generation

As mentioned in section 3.1, the ray generation algorithm is important for the overall behaviour of the algorithm. In particular, the distribution of rays in update traces should be such that the total distribution of rays, for the combination of all traces, matches the target distribution. If a random distribution is used, it is sufficient to ensure that the same distribution is used for sampling every trace. The normalization process should ensure that the energy distribution remains constant.

If a deterministic grid is used, things become more complicated. In order to maintain proper energy distribution, the distribution of all ray generation points should be considered for each updated trace.

5 PREDICTED ISSUES

While some research in the area of human attention, and some of the practical aspects of raytracing algorithms suggest that this can be a useful and efficient idea, some problems can be foreseen already.

One relates to the behaviour when the agent is moving very slowly through the environment. As long as the agent has not moved too far from the initial impulse response, this should lead to an increase in simulation quality over time. However, at some point the response should be regenerated, as specified by the spatial resolution, and this could lead to a sudden and perceptible decrease in quality. The other option is that the response should only be updated, not reset. While the second option would not lead to a sudden change in sound and sound quality, it could lead to a drift where the impulse response no longer corresponds to the current location in the environment. Issues such as these could possibly be solved using some sort of moving average process, like what was discussed in section 4.3, so that the older samples are slowly becoming less and less relevant. If this should be efficient and still provide the advantages of the algorithm presented in this paper, however, there is a fine balance to strike regarding about how long each trace should remain relevant. This is a promising path forward, but more research is needed into how this may be implemented.

6 FURTHER RESEARCH

As this paper presents an algorithm still in its earliest phases, a lot of research remains. As a first step, the algorithm should be implemented. While some possible issues and some peculiarities are discussed in this paper, it is likely that more will be uncovered as the work progresses. Presenting the idea to the acoustic community is also expected to introduce more questions, and more areas of research.

In a first step, subjective small-scale testing should provide some indications of how well the system works, and in determining what would be reasonable values for temporal and spatial constants as discussed in section 4.

In this paper, only a brief discussion regarding the possibility of implementing some sort of moving average method is discussed. This could, however, provide a promising future research avenue, and could offer a more robust performance to what is suggested in this paper.

REFERENCES

[1] Use spatial audio in 360-degree and VR videos. https://support.google.com/youtube/answer/6395969?hl=en-GB, May 2019.

[2] R. Eramudugolla, D. R. F. Irvine, K. I. McAnally, R. L. Martin, and J. B. Mattingley. Directed attention eliminats 'change deafness' in complex auditory scenes. *Current Biology*, 15(12):1108 – 1113, 2005.

[3] S. Pelzer, D. Schröder, and M. Vorländer. The number of necessary rays in geometrically based simulations using the diffuse rain technique. *Fortschritte der Akustik - DAGA*, pages 323–324, 2011.

[4] A. Rimell and A. Owen. The effect of focused attention on audio-visual quality perception with applications in multi-modal codec design. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Precessing. Proceedings*, 2000.