Tree Traversal Algorithms for Real Time Sound Propagation Calculation

Panagiotis Charalampous[1], Despina Michael[1]

[1]*Games and Emerging Technologies Lab, Cyprus University of Technology, Limassol, Cyprus*

Correspondence should be addressed to Panagiotis Charalampous-Author (pc.charalampous@edu.cut.ac.cy)

ABSTRACT

A major category of algorithms for simulating sound propagation in three dimensional environments is the category of geometrical acoustics which are based on various tracing techniques. All tracing techniques are essentially tree traversals throughout a tree of candidate sound paths which could potentially affect the sound field at a given receiver location. Even though there is significant literature for speeding up tracing in three dimensional spaces for use in real time applications, the type of these algorithms has barely been discussed. In this work, we implement an image source method variant and we compare three different tree traversal approaches, depth-first, breadth-first and best-first. We examine their performance in respect to time and we highlight the advantages and disadvantages of each one and evaluate the applicability of each in real time sound propagation calculations.

1. INTRODUCTION

Algorithms for fast calculation of sound propagation is a hot research topic both in audio and computer graphics field. Fast and accurate sound rendering is an important component of an interactive virtual environment [1] [2]. Sound serves significant functions in such environments such as delivering important information, attracting user's attention, assisting in user's orientation, evoking emotional responses, creating ambience or suggestion, creating anticipation for future events, resonating memorable moments, enhancing the environment's structure and adding sense of location. But sound rendering calculations are computationally expensive. Therefore improved calculation times for accurate acoustics calculations are required by commercial acoustics software applications, as current algorithms are often complicated and time consuming [3]. As a result, research has been focused on speeding up the calculation of sound propagation through space [4].

There are three major categories of methods for sound simulation. These are a) Geometrical propagation methods b) Numerical methods and c) Hybrid methods. In brief, in geometrical propagation methods sound is described as a ray phenomenon [3]. Thus sound rays propagated within an environment are used for the estimation of a sound field at a given receiver position. Tracing techniques are those techniques used in geometrical acoustics that detect the propagation of such sound paths in a three dimensional space. Numerical methods are methods which subdivide the space or boundaries to smaller elements and solve progressively the wave equation in order to estimate the sound field of the space [5]. Hybrid methods are combinations of the two aforementioned categories.

In this paper, we will focus on geometrical propagation methods. Most geometrical propagation algorithms can be described as tree traversals. It is widely known in computer science that a tree can be traversed in many different ways with each different type of traversal affecting the algorithm's performance. As a result, tree traversals are a major topic in the field of artificial intelligence [6].

2. RELATED WORK

Geometrical acoustics is the category of methods used in acoustics to describe sound as a ray phenomenon. There are various subcategories of algorithms under the geometrical acoustics definition based on the tracing technique used. These are the image source methods [7] [8] [9] [10] [11], beam tracing [12] [13] [14] [15] [16] [17] [18], frustum tracing [19] [20] [21], ray tracing [22] [23] [24] [25] [26] [27] and particle tracing[28] [29]. These methods share a common concept, that of tracing sound rays emitted by a sound through an environment until they reach a receiver, and then calculating the expected impulse response at that receiver.

Image source method is the method of finding virtual sources by considering each polygonal surface in the environment as a reflector. Virtual source locations are assumed as new locations of sound sources and are used in a recursive way to calculate the propagation of sound in space. Image source method is a method that provides accurate results, as it detects all the possible sound reflections in a three dimensional environment, but suffers from poor performance. Beam tracing is a method of tracing polyhedral beams in space and then casting them to rays. In beam tracing, beams are cast throughout the environment. Each beam is intersected with each polygon in the environment in a front to back order. Then, the beam is clipped removing the shadow region created by the polygon. After that, the shadow region is used to construct a transmission beam and image source method is used to construct the reflection beam. The final rays that arrive at the receiver are used for the computation of the impulse response.

Ray tracing is used as a technique for generating an impulse response by tracing the path of a sound ray throughout a three dimensional environment and calculating the various effects that occur when that ray encounters an obstacle. The disadvantages of ray tracing techniques are that the discrete number of rays traced and the arbitrary shape of rooms might lead to significant loss of paths or paths counted multiple times [25]. Particle tracing is a variation of the ray tracing technique and it is usually referred as phonon tracing or sonel mapping. Phonon tracing is inspired by the photon mapping technique used in computer graphics rendering. Frustum tracing is an approach similar to beam tracing, with the difference that it performs an approximate clipping by subdividing the beams to sub-frusta, opposed to beam tracing which performs accurate clipping of beams. Frustum tracing combines the efficiency of interactive ray tracing with the accuracy of tracing a volumetric representation.

All the aforementioned techniques used are essentially tree traversal algorithms. However, to the best of our knowledge, there is no research that relates geometrical acoustics techniques with tree search problems, which are studied within the artificial intelligence domain. In the following section, we describe why these techniques can take the form of a tree search and we describe three basic algorithms of searching tree data structures.

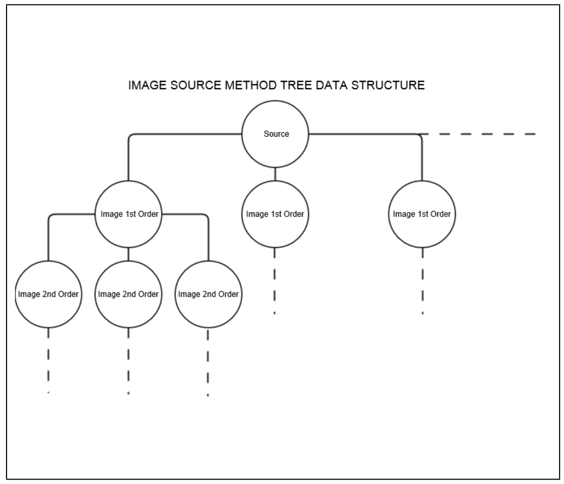

Fig. 1: Representation of the tree structure of an image source algorithm.

3. TREE TRAVERSALS IN GEOMETRICAL ACOUSTICS

As mentioned above, these techniques can be represented in computational terms as a tree data structure. The concept is similar in all the aforementioned techniques. In the case of the image source method, an image of the source is created for each reflecting surface, forming this way the first level of nodes in the tree. Following on, all images in the first level of the tree are used to create second order images, representing second order reflections. This process continues until an interruption criterion is met, such as the maximum depth of the tree, the distance of an image source to the receiver or any other criterion set by the algorithm (see Fig. 1).The sound at the listener is computed based on the reflections derived by the tree traversal. In a similar way, all other tracing techniques can be represented as trees. Beams and frusta intersect with polygons, which then become new nodes that shoot new beams and frusta in a recursive way until they reach the receiver or an interruption criterion is met. The same logic applies for ray tracing and particle tracing algorithms.

The tracing problem can be represented in the form of a tree traversal problem. This fact implies that the solution can be found using a tree traversal algorithm. Tree traversal algorithms are widely studied in the domain of artificial intelligence and a lot of variants exist but they are mainly divided into two major categories, depth-first tree traversals and breadth-first tree traversals [6].

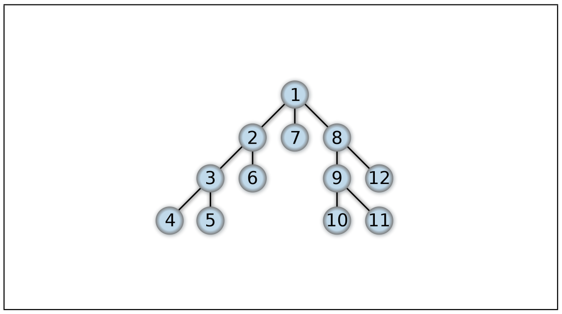

Fig. 2: Representation of a depth-first preorder traversal.

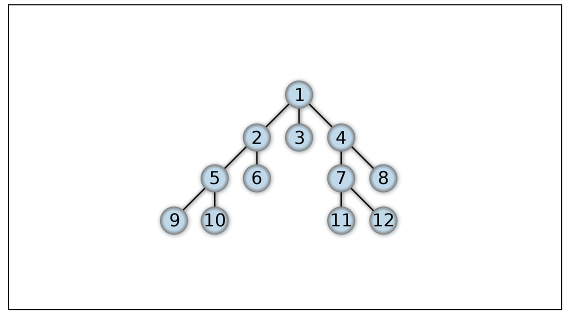

Fig. 3: Representation of a breadth-first traversal.

In the case of geometrical acoustics, the tree to be traversed can be either constructed during tracing or preconstructed before tracing. This decision has important implications in the algorithm's performance. Preconstructed trees require memory allocation for the entire tree and predefined termination criteria like the depth of the tree. Also preconstructed trees imply static geometries. On the other hand dynamically constructed trees require less memory and allow dynamically set termination criteria but the dynamic creation of the nodes results to a performance overhead in the case the calculation is repeated many times for the same tree. Following on, a more detailed description will be given for each category, as well as for a subcategory of breadth-first variants, called best-first approaches.

3.1. Depth-first Tree Traversal

A depth-first tree traversal algorithm is an algorithm where the nodes of the tree are expanded in depth as far as possible e.g. until an interruption criterion is met, before the algorithm backtracks to the previous nodes [31]. Depth-first algorithms come in many types like preorder, postorder and inorder, defined by the order that the root is examined related to the left and right subtree. The most common type used in geometrical acoustics is the preorder depth-first approach. Figure 1 demonstrates the order in which the nodes of the tree will be examined in the case of a depth-first preorder traversal.

The main benefit of a depth-first algorithm occurs in the case of a dynamic tree construction. When a tree is constructed dynamically, implying that child nodes are added only after a parent node has been visited, then the memory requirements of this algorithm are minimal, as the only traversal related information needed to be stored in memory at any time are the parent nodes of the visited node at each moment. This makes a depth-first traversal memory efficient as well as fast because no search queue needs to be maintained. However, the main disadvantage of depth-first is need of an a priori set termination criterion. Without a termination criterion, the tree will expand forever on the leftmost path of the tree. As a consequence, the programmer or the user needs to set beforehand a termination criterion for the algorithm. This is a drawback for a real time approach as we will show later on in this paper.

3.2. Breadth-first Tree Traversal

In breadth-first tree traversal, the algorithm searches the tree in the following way. Starting, it first visits the root node. Then it finds all the child nodes and adds them in a queue to be examined. Following on, it recursively takes a node from the queue, examines it and adds all its child nodes in the queue. This process results into an order of visiting nodes as the diagram shown in Figure 3. The main benefit of a breadth-first algorithm is that it can traverse all the width of the tree without an explicit a priori set termination criterion. Contrary to a depth-first algorithm, a breadth-first one can traverse an infinite tree in infinite time without a termination criterion. On the other hand, a major drawback of breadth-first search is the management of the queue. As the depth of the search deepens, the queue gets larger exponentially, making the algorithm memory inefficient and unpractical for searches requiring traversals in tall trees.

3.3. Best-first Tree Traversal

Best first tree traversal is a mutation of a breadth-first tree traversal. It follows the same principles as breadth-first, except that the queue is a priority queue. In a priority queue, nodes are not dequeued in First In First Out(FIFO) order but based on a priority function. The outcome of the priority function depends on the node that it evaluates, the problem to be solved, the information gathered by the search up to that point on any extra knowledge we might have about the problem[6]. This way the algorithm expands each time the most promising node based on a specified rule.

3.4. Tree Traversals in Current Techniques

As described in section 2, there is a variety of techniques for geometrical sound propagation which are based on a tree traversal approach. We examined some of the implementations in order to determine the type of tree traversal used in each case. Notably, only some mention explicitly the method used and provide justification for their choice [8] [12] [13] [16] [18]. [8] uses a depth-first approach for memory reasons [12] uses depth-first without any further justification and [13], [15] and [16] use best-first in comparison with a depth-first approach. [18] use breadth-first for multi-threading purposes. Other implementations do not mention the approach used but it can be extracted by examining the algorithm's description. [10] and [9] use breadth-first approaches and [19] [20] and [21] seem to be using a depth-first approach.

A notable conclusion from this short survey is that except [15] and [16], where performance comparisons take place, no other implementation justifies the traversal type choice based on performance criteria. On the other hand, it seems that the choice is usually related to implementation convenience. Hence, in the following sections we validate three traversal types based on their performance in respect to time and we discuss the outcomes.

4. VALIDATION OF TREE TRAVERSALS

We have implemented an image source algorithm for the detection of specular reflections in three variants, for each respective traversal as described above, to evaluate the qualitative differences of the three different types. Our image source algorithm uses a form of visibility tracing which assists in pruning the image tree by eliminating invalid image nodes from non visible trees. The tree is constructed dynamically at run time. Thus, we avoid preconstructed trees that impose a priori boundaries to the search process. We ran the algorithms on four configurations with different geometrical characteristics and we compare the results. The aim of the comparison was to investigate how these algorithms compare in detecting valid images over time and how the traversal type affects the outcome. We decided to make this type of comparison to evaluate the different approaches in relation to real time requirements where time constrains can affect significantly the outcome of the search process.

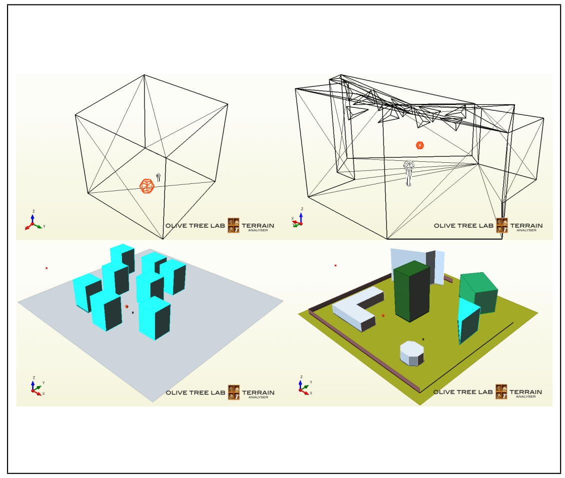

Fig. 4: Configurations. Top Left: Cubic Room. Top Right: Choir Room. Bottom Left: Urban Environment. Bottom Right: Outdoor Environment

We chose four different geometrical configurations, aiming at representing the best way possible typical scenarios where real time sound propagation calculations could take place. The algorithms have been implemented using C# and the PEMARD AcousticsLib [30]. Following on we present the results for each case.

For the depth-first approaches we have set a termination criterion based on the maximum reflection order, as this is a requirement for a sensible execution of the algorithm, as described above. In order to compensate for the need to set a maximum reflection order in the depth-first case, we ran the algorithm with three different maximum orders and compared the outcomes with the other two cases in order to evaluate how this decision affects the performance. Then, for breadth-first and best-first search we took the number of valid images detected by the depth-first search with the highest termination order and set it as a termination criterion.

4.1. Priority Function

For the case of the best-first algorithm we had to use a priority function that would guide the algorithm. We decided to use a heuristic priority function based on information gathered from the parent nodes of each evaluated node. As a result, the priority function used is the following.

$$f(n) = 10VAi + Pio$$

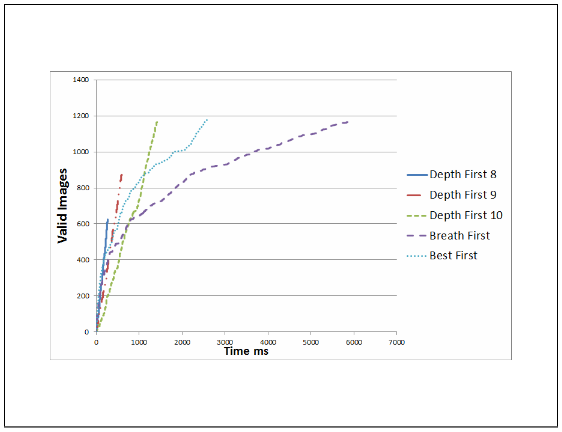

Fig. 5: Cubic Room.

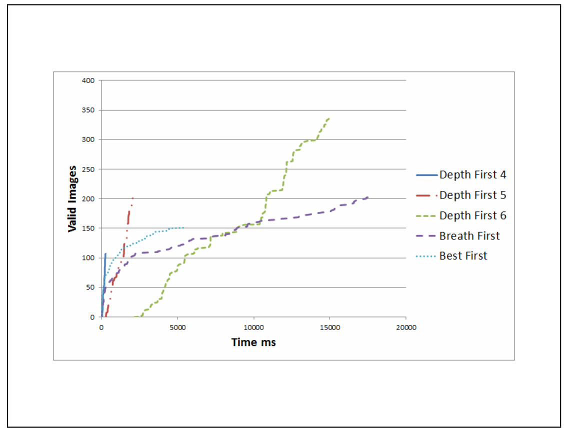

Fig. 6: Choir Rehearsal Room.

where VAi (Valid Ancestor images) is the number of nodes that are ancestors to the node under investigation, that created valid reflections to the receiver and Pio(Parent image order) is the reflection order of the parent node. We consider VAi as a good indicator of the importance of the current node because the more valid parents a node has, it is more likely that the node will be valid too. The factor of 10 is added so we can add the parent image order as a last digit to be used in the case of equality. For example, if for two nodes the VAi equals to 3 and the one represents a 6th order image and the other a 7th order image, then the node at the lowest order will have a priority of 35 while the highest order one will have a priority of 36. Thus, the 6th order node will be assigned priority. This priority function has been selected for its ease of implementation and from empirical observations that it provides a decent prioritization without implying that this is the best possible priority function.

4.2. Cubic Room

The first model under investigation is a cubic room with dimensions of 10m X 10m X 10m. We choose to start with a cube as the simplest common geometrical structure resembling an indoor situation. The lowest bottom corner at(0,0,0) and the opposite one at (10,10,10). We set the source at (4,5,1) and the receiver at (6,5,1). The maximum reflection order was 8, 9 and 10 for the depth-first searches. The maximum number of valid images for breadth-first and best-first was 1180. Figure 5 presents the results obtained for the three depth-first searches, the breadth-first and the best-first search. We observe that depth-first searches have a linear pattern in the increase of valid image sources per time. On the other hand, in the case of breadth-first and best-first searches, we note that the rate of increase slows down over time. At this stage, it is important to highlight the difference in performance of the three depth-first variants. A change in the maximum order termination criterion can change the performance of the algorithm by a great amount when comparing the valid image sources detected over time. We will provide a detailed explanation of this phenomenon in the discussion section.

When it comes to comparing best-first and breadth-first searches, as it would be intuitively expected, best-first search outperforms breadth-first search. This is because best-first search is guided by the priority function. The general delay to reach the same number of images as the depth-first search is attributed to the management of the queue required in both cases.

4.3. Choir Room

The next space under investigation is a choir rehearsal room, as seen in Figure 4. The searches have been run with the same mindset as in the cubic room. The termination orders for the depth-first searches were 4, 5 and 6 orders respectively. The maximum paths for the breadth-first and the best-first searches were 337. In the case of the choir room, we can discern similar patterns as in the cubic room case. depth-first searches grow in a rather linear way and the performance is affected by the termination criteria while the performance of best-first and breadth-first deteriorates over time. Best first still outperforms the breadth-first algorithm.

4.4. Urban Environment

In the case of the urban environment, we choose a model

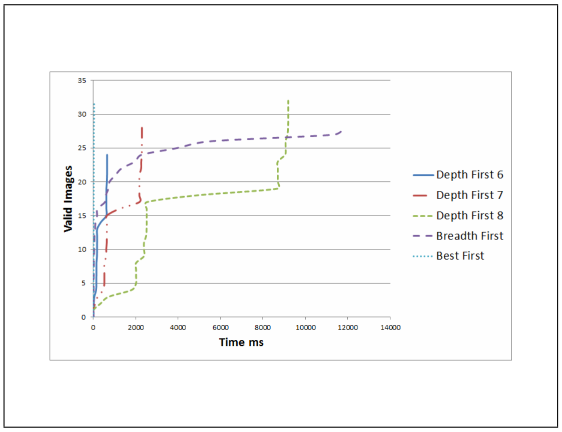

Fig. 7: Urban Environment.

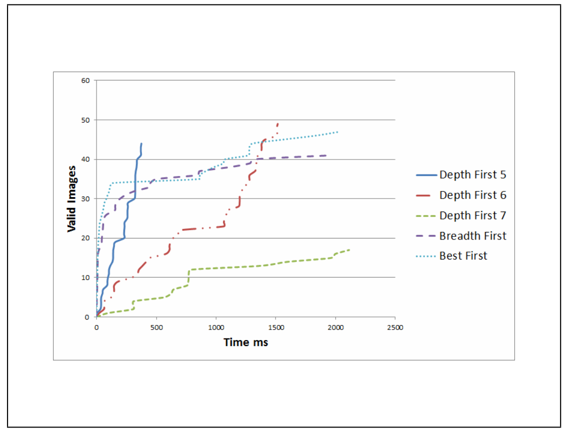

Fig. 8: Outdoor Environment.

that carries typical characteristics of environments found in urban topologies. More specifically, in a modern urban setting, we can observe many cuboid shaped structures aligned almost in parallel, with a lot of faces facing each other. Figure 4 shows the model used for this scenario. The termination orders for the depth-first searches where 6, 7 and 8 orders respectively. The maximum paths for the breadth-first and the best-first searches were 32.

In this case, we see the patterns of behavior of the search algorithms change. The most important change is the rapid detection of valid image sources by the best-first approach. This is attributed exclusively to the priority function used. Environments of this type have many faces with nearly opposite normal directions. This contributes to the creation of repetitive reflections between these surfaces up to high orders. The priority function described earlier benefits from such occurrences and can detect very fast these repetitive reflections. As a result, we can see the explosion of the images detected by the best-first search in Figure 7. Other important remarks are the sharper growth of breadth-first during the first milliseconds and the sharper slowdown for the rest of the execution as well as the non linear behavior of depth-first searches and the dramatic slowdown in the case of setting the termination order to 8.

4.5. Outdoor Environment

The outdoor environment is an environment where various buildings are distributed through the space in a rather random way. This type of environment could resemble an industrial site or an army camp. The termination orders for the depth-first searches were 5, 6 and 7 orders respectively. The maximum paths for the breadth-first and the best-first searches were 56.

In this case, we can again see the following a) the rapid increase of both best-first and breadth-first algorithms during the first milliseconds of execution, b) the rapid decrease in their performance after a certain point c) the fact that best-first is outperforming breadth-first and d) the fact that depth-first searches are highly influenced by the highest order termination criterion.

5. DISCUSSION

The main outcome of this research is that different traversal algorithms have different behavior patterns and this fact can be a catalytic factor for their applicability in different scenarios. In this section, we will discuss the algorithms used in this study by highlighting benefits and advantages of each algorithm as well as the cons and disadvantages, as these have been concluded by the test cases described above.

Beginning with the depth-first search, we can point out its advantages which are a) simplicity in implementation b) memory efficiency c) the fastest traversal of the entire tree. Depth-first is simpler in implementation that the other two approaches as it requires only a recursive function without the need to maintain a queue. This fact also makes it memory efficient. This also results in a faster overall traversal of the entire tree, when compared to the other approaches. On the other hand, the main disadvantages are the following a) it requires a maximum depth to be set beforehand b) the rate of detecting valid image sources in the starting milliseconds of the algorithm is not as high as the other two. This can be explained by the fact that the ratio of valid images over the total number of images is higher at the top of the tree rather at the bottom while this doesn't hold true when you examine the distribution of valid sources from left to right. Thus breadth-first and best-first have this characteristic shape of curve, rapidly increasing in the first milliseconds while slowing down at the later stages, in contrast to depth-first which has a linear increase.

Based on the highlighted advantages and disadvantages, the depth-first algorithm is ideal for scenarios where a tree needs to be traversed up to a specified depth known beforehand and adequate time is available. Such a scenario could be for example software applications for room acoustics analysis. On the contrary, depth-first is not ideal for real time applications such as video games and virtual reality. This is because in the case of real time scenarios very strict time constrains exist and performance over time is an important component. In this case depth-first suffers from two main weaknesses a) since a highest termination criterion needs to be set from the start, the algorithm can perform faster if a low cut off order is set or perform much slower if a high order is set (See for example Figure 8.). b) If the algorithm does not manage to search the entire tree withing the real time constrain, it might lose important low images, since the traversal examines the tree in a left to right order rather than in a bottom down manner.

The advantages and disadvantages of the depth-first approach can deductively lead us to the advantages and disadvantages of the other two approaches, breadth-first and best-first. We can see that even though memory requirements and the excessive slow down over time make these approaches inappropriate for long running in depth searches, it makes them ideal for real time sound rendering applications. This is because these approaches, having a top down approach in searching the tree, can detect the most important valid nodes, which are usually gathered in the higher part of the tree, in the first milliseconds of the search process. The top down approach is an inherent prioritization of the image sources within the structure of the tree and since prioritization is of major importance in real time applications, they benefit by such an approach. Moreover, the higher density of valid nodes in the top parts of the tree provides a very good performance benefit to these algorithms during the initial time period of execution. Finally, the absence of a required termination order criterion removes the performance dependence from user decisions.

Furthermore, when the comparison is done between a best-first approach and a breadth-first approach, the use of a priority function offers an obvious benefit. Nodes that can be heuristically evaluated as more important are expanded before others, providing enhanced performance over time when compared to a simple breadth-first implementation. This benefit is vivid in all four comparisons we performed in this case study. As a result, it can be concluded that from the three algorithms compared, the best-first approach seems to have the best behavior when it comes to real time sound rendering requirements.

6. CONCLUSION AND FURTHER WORK

Concluding, in this paper we made a comparison between three different tree traversal algorithms, as these would be implemented in a typical image source approach, in four different geometries with different characteristics. We compared the results and we have highlighted on the advantages and disadvantages of each approach. We have shown that while depth-first approach is an ideal approach for time consuming in depth acoustics calculations, when the case is an application with real time constrains, breadth-first variants have significant benefits. On top of that, approaches that are guided by priority functions offer improved performance when compared to approaches without any prioritization.

The concept of priority driven tracing has been investigated in the past by Funkhouser and Min [13] [15] for the case of a beam tracing algorithm with interesting outcomes, but no further work has been carried out since then, at least none to the knowledge of the authors. With this paper we have established that best-first tree traversal seems to perform better for real time approaches when compared to other tree traversal algorithms. As a result, further work will include investigation of the performance of various priority functions used in such algorithms and extension of the concept to other acoustical phenomena like sound diffractions. Also, we will evaluate the performance over more parameters like reverberation time, total energy, instructions required, memory requirements and other related parameters.

7. REFERENCES

[1] S. Huiberts, "Captivating sound the role of audio for immersion in computer games". 2010. University of Portsmouth.

[2] P. Peerdeman, "Sound and Music in Games" 2009, And Haugehåtveit

[3] P. Charalampous, P. Economou, ”A Framework for the Development of Accurate Acoustic Calculations for Games”. Audio Engineering Society Conference: 49th International Conference: Audio for Games. 2013 February. Audio Engineering Society.

[4] D. Manocha and M. C. Lin, ”Interactive sound rendering”. Computer-Aided Design and Computer Graphics, 2009, 19-26, IEEE

[5] M. Vorlander, ”Auralization: fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality”. 2010, Springer Publishing Company, Incorporated

[6] R.D. Russell, J. Stuart, et al. Artificial intelligence: a modern approach. Vol. 2. Englewood Cliffs: Prentice hall, 1995.

[7] J. Allen and D.A. Berkley, ”Image method for efficiently simulating small-room acoustics”. The Journal of the Acoustical Society of America, 1979, 65, 943

[8] J. Borish, ”Extension of the image model to arbitrary polyhedra”. The Journal of the Acoustical Society of America, 1984, 75, 1827

[9] F.P. Mechel, ”Improved mirror source method in roomacoustics”. Journal of sound and vibration, 2002, 256, 873-940

[10] M. Vorlander, ”Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm”. The Journal of the Acoustical Society of America, 1989, 86, 172

[11] D. Schröder and T. Lentz, ”Real-time processing of image sources using binary space partitioning”. Journal of the Audio Engineering Society, 2006, 54, 204-619

[12] Funkhouser, Thomas, et al, ”A beam tracing approach to acoustic modeling for interactive virtual environments.”, Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM, 1998.

[13] T. Funkhouser, P. Min, I. Carlbom, ”Real-time acoustic modeling for distributed virtual environments”. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 1999, ACM Press/Addison-Wesley Publishing Co., pp. 365374

[14] F. Antonacci, M. Foco, A. Sarti, S. Tubaro, ”Fast modeling of acoustic reflections and diffraction in complex environments using visibility diagrams”, In Proceedings of 12th European signal processing conference, 2004, pp. 17731776

[15] P. Min,T. Funkhouser, ”Priority-driven acoustic modeling for virtual environments”, In Computer Graphics Forum 2000, vol. 19, Wiley Online Library, pp. 179188

[16] T. Funkhouser, N. Tsingos, I Carlbom, G. Elko, M. Sondhi, J.E. West, G. Pingali, P. Min, A. Ngan: ”A beam tracing method for interactive architectural acoustics”, The Journal of the Acoustical Society of America 115, 2004, 739.

[17] S. Laine, S. Siltanen, T. Lokki, L. Savioja: ”Accelerated beam tracing algorithm”, Applied Acoustics 70, 1 2009, 172181

[18] M. Sikora, I. Mateljan: ”A method for speeding up beam-tracing simulation using thread-level parallelization”, Engineering with Computers 2013, 110

[19] C. Lauterbach, A. Chandak, D. Manocha, ”Interactive sound rendering in complex and dynamic scenes using frustum tracing” Visualization and Computer Graphics, IEEE Transactions on 13, 6, 2007, 16721679

[20] A. Chandak, C. Lauterbach, M. Taylor, Z. Ren, D. Manocha, ”Ad-frustum: Adaptive frustum tracing for interactive sound propagation”, Visualization and Computer Graphics, IEEE Transactions on 14, 6, 2008, 1707172

[21] M. Taylor, A. Chandak, Z. Ren, C. Lauterbach, D. Manocha, ”Fast edge-diffraction for sound propagation in complex virtual environments”, In EAA auralization symposium 2009, pp. 1517

[22] J. C. Allred, A. Newhouse, ”Applications of the monte carlo method to architectural acoustics”, The Journal of the Acoustical Society of America 30 ,1958

[23] A. Krokstad, S. Strom, S. Sørsdal, ”Calculating the acoustical room response by the use of a ray tracing technique”, Journal of Sound and Vibration 8, 1, 1968, 118125

[24] A. Kulowski: ”Algorithmic representation of the ray tracing technique”, Applied Acoustics 18, 6, 1985, 449469

[25] H. Lehnert, ”Systematic errors of the ray-tracing algorithm. Applied Acoustics”. 38, 2, 1993, 207221

[26] M. Taylor, A. Chandak, Q. Mo, C. Lauterbach, C. Schissler, D. Manocha: ”Guided multiview ray tracing for fast auralization”, 2012, IEEE

[27] M. Dreher, G. Dutileux, F. Junker, et al, Optimized 3d ray tracing algorithm for environmental acoustic studies. Acoustics 2012 Nantes, 2012

[28] M. Bertram, E. Deines, J. Mohring, J. Jegorovs, H. Hagen, ”Phonon tracing for auralization and visualization of sound” In Visualization, 2005, VIS 05. IEEE (2005), IEEE, pp. 151158.

[29] B. Kapralos, ”The sonel mapping acoustical modeling method” PhD thesis, York University Toronto, Ontario, 2006.

[30] P. Economou, P. Charalampous, G. Amadasi, ”Development of Accurate and Fast Acoustic Calculations using Olive Tree Lab-Acoustics Lib.”, AIA-DAGA Conference on Acoustics, 2013.

[31] D. E. Knuth. ”The art of computer programming.” Fundamental Algorithms 1999.