

Fast Evaluation of the Room Transfer Function Using Multipole Expansion

Ramani Duraiswami, *Member, IEEE*, Dmitry N. Zotkin, and Nail A. Gumerov

Abstract—Reverberation in rooms is often simulated with the image method due to Allen and Berkley (1979). This method has an asymptotic complexity that is cubic in terms of the simulated reverberation length. When employed in the frequency domain, it is relatively computationally expensive if there are many receivers in the room or if the source or receiver positions are changing with time. The computational complexity of the image method is due to the repeated summation of the fields generated by a large number of image sources. In this paper, a fast method to perform such summations is presented. The method is based on multipole expansion of the monopole source potential. For offline computation of the room transfer function for N image sources and M receiver points, use of the Allen–Berkley algorithm requires $O(NM)$ operations, whereas use of the proposed method requires only $O(N + M)$ operations, resulting in significantly faster computation of reverberant sound fields. The proposed method also has a considerable speed advantage in situations where the room transfer function must be rapidly updated online in response to source/receiver location changes. Simulation results are presented, and algorithm accuracy, speed, and implementation details are discussed. For problems that require frequency-domain computations, the algorithm is found to generate sound fields identical to the ones obtained with the frequency-domain version of the Allen–Berkley algorithm at a fraction of computational cost.

Index Terms—Acoustic fields, architectural acoustics, array signal processing, multipole expansions.

I. INTRODUCTION

IN THE TYPICAL inhabited environment, the sound reaching a receiver is not an exact copy of the sound emitted by the transmitter. The received sound is scattered off various surfaces in the environment as well as off the transmitter and off the receiver (if they have finite extent). The collective effect of the sound interaction with the room walls and contents is known as reverberation. Formal studies of reverberation date back to 1900 [2]. Since then, numerous studies on how reverberation affects human perception of sound and music have been conducted. Several distinct research directions can be noted, such as optimization of the room reverberant response for perceptually pleasant listening experience, for the best performance of a human subject doing a specific auditory task (such as localization of an audio source in the environment

or speech comprehension), and for the best performance of a certain computer algorithm (such as localization of an active sound source and enhancement of its signal). The first of these three directions has evolved into a separate branch of acoustics called architectural acoustics (e.g., [3], [4]). Example studies for the second and the third directions mentioned are [5], [6], and [7], [8], respectively.

The reverberation properties of a room play an extremely important role in determining the listening experience. For example, in architectural acoustics there exist certain principles for constructing performance spaces to assure that the room acoustics helps the perception of the performance rather than ruining it. In virtual acoustics and audiovisual environments simulation of reverberation is necessary to assist with externalization of sounds, to achieve a sense of presence, and to convey information about the virtual environment properties (room size, surface materials, etc.) In teleconferencing it is necessary to account for the reverberant properties of the room to compensate for the acoustic feedback and to improve the accuracy of localization of an active speaker (e.g., for pointing a pan-tilt camera at the speaker). Finally, enhanced spatially selective sound capture can be performed with knowledge of the sound source position and room geometry by focusing the acoustic capture unit not only on the main acoustic source but also on its coherent reflections in the walls of the room [a technique known as matched filter array (MFA) processing].

In many studies that involve the development of various audio algorithms, it is necessary to perform validating tests under reverberant conditions. Sometimes such tests are done in an actual experimental room. However, as the reverberant response of a room is a function of its geometry, of the source and the receiver locations, of impedance characteristics of the room walls, and of several other parameters, it might be an expensive project to perform a comprehensive experimental evaluation. A more economical, and usually employed, alternative is to synthesize the reverberant response for various rooms using computer models. As such, methods for modeling room reverberation have many uses in various application areas where prediction of a room response to an arbitrary stimulus is necessary.

One of the first artificial reverberators aimed at reproducing the perceptual characteristics of reverberation was developed by Schroeder [9] and later greatly improved by Moorer [10]. These early methods did not attempt to model sound propagation and were targeted only at the creation of a perception similar to that obtained in a real room. The first geometric model of room reverberation is due to Allen and Berkley [11]. This paper became the basis for many subsequent papers (e.g., [12]–[15]) and is heavily cited in journals and conferences, including in microphone array studies [16], [17], speech processing [18], [19],

Manuscript received January 5, 2005; revised December 21, 2005. Some of the results in this paper were presented in abbreviated form in [1]. This work was supported in part by the National Science Foundation under Award 0086075 and Award 0219681. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rainer Martin.

The authors are with the Perceptual Interfaces and Reality Laboratory, UMIACS, University of Maryland, College Park, MD 20742 USA (e-mail: ramani@umiacs.umd.edu; dz@umiacs.umd.edu; gumerov@umiacs.umd.edu).

Digital Object Identifier 10.1109/TASL.2006.876753

and virtual auditory environments [20]–[22]. The basic idea of the Allen–Berkley model is that the reverberation can be represented as the effect of an infinite number of image sources that are created by reflecting the true acoustic source in room walls. The walls are assumed to be specular sound reflectors, the wall reflection coefficients are real and frequency-independent, and the room has a simple rectangular shape, which greatly simplifies the computation of the image source positions. Under these conditions, the simulated room impulse response (RIR) is obtained by a simple time-domain summation of the contributions of each image source with appropriate strengths. The Fourier transform of the RIR is commonly called the room transfer function (RTF). The model was extended in [12] to the case of arbitrary piecewise-planar rooms and in [14] to the case of directional and/or shadowing source and/or receiver. Additionally, it was noted that it is generally unnecessary to model the reverberation as accurately as possible beyond a certain time (about 80 ms) after the arrival of the direct sound [23], [24], and consequently approximations of a statistical nature aimed at evoking the appropriate perception are usually employed to model later parts of the reverberation (e.g., [9], [25], [26]).

As mentioned before, the Allen–Berkley model operates in the time domain with significant simplifying assumptions. It is known that not for all rooms is the reverberation time frequency-independent [14] because the reflective properties of materials often vary with frequency [27]. In addition, sound absorption by the air is frequency-dependent [28]. The desire to extend the model to perform simulations separately in different frequency bands is expressed in several papers [13], [14], [29], and in some more recent papers [6], [30] such fine-band RTF modeling is actually employed. As a particular example, MFA processing [31]–[33] relies on fast RIR computation in an attempt to coherently pick up not only the source signal itself but also its significant reflections by filtering the received signal with the time reverse of the RIR for given source and receiver positions, resulting in a significant output signal SNR increase compared to simple beamforming. The image model [11] is used to model the RIR in [33]. Application of MFA to real scenarios (such as teleconferencing) calls for a dynamic RIR update in response to speaker motion, and while the time-domain image model is sufficiently fast to enable real-time implementation, it cannot account for frequency-dependent wall reflection coefficients, which could significantly distort the RIR and consequently the beamformed signal.

There exist two distinct scenarios for computation of the room transfer function. In the first (offline) case, the acoustic scene is static and may include many acoustic sources, and the RTF is to be computed for many receiver locations. Direct application of the Allen–Berkley algorithm in this case results in a computational complexity that grows as the product of the number of sources and the number of receivers. In the second (online) case, it is necessary to dynamically update the RTF for a fixed location of the acoustic source and changing receiver location with the update latency being the critical performance measure (i.e., any time spent in computations that are independent of the receiver location and therefore can be done in advance is ignored). The problem can also be posed for a fixed receiver (e.g., a microphone in the microphone array) and a moving source by using the reciprocity principle [34]. The goal of current paper

is to introduce a fast, frequency-domain, multiple-receiver RTF computation algorithm based on multipole expansion that accelerates computation in both of these situations, with greater benefits in the online case.

The algorithm is based on exploitation of the fact that the Allen–Berkley expression for the RTF solution is nothing but the sum of the potentials of many monopole sources and can be factorized into receiver-location-dependent and receiver-location-independent parts via the multipole translation identity. Similar factorizations are also used in the fast multipole method, which was introduced first in [35] and is now widely used to accelerate computations for different problems, including the boundary element method [36], [37], multiple scattering from many objects [38], and others. The algorithm can achieve several orders of magnitude improvement in computational complexity if the frequency-domain RTF is needed. If the total number of (actual and image) sources is N and the RTF is to be computed at M points, then straightforward RTF computation with the image model [11] requires $O(NM)$ time, whereas RTF computation with the proposed method can be done in $O(N + M)$ time by initially assembling the vast majority of the image sources into one complex multipole source located in the center of the room (the assembly process does not require knowledge of the position of evaluation points) and then evaluating the potential of that multipole source at the evaluation points. Furthermore, the computational complexity of the multipole source potential evaluation is directly proportional to the size of the room in which the multipole source is used to represent the sum of potentials of image sources; therefore, the RTF evaluation in the online scenario can be done arbitrarily fast with the help of the room subdivision technique covered in Section VI of this paper.

The rest of the paper is organized as follows. In Section II, a brief overview of the problem is given and the Allen–Berkley algorithm [11] is discussed. In Section III, the proposed multipole algorithm is described. In Section IV, the error analysis is given and guidelines for the choice of the implementation parameters are presented. In Section V, the computational complexity of the Allen–Berkley algorithm and the proposed algorithm is analyzed. In Section VI, a modification of the multipole algorithm that is suitable for real-time implementation as well as for raising the high frequency limit is proposed. In Section VII, comparisons between the results produced by the Allen–Berkley method and by the proposed method are shown. In Section VIII, performance and implementation considerations are discussed. Finally, Section IX concludes the paper.

II. FORMULATION AND THE ALLEN–BERKLEY ALGORITHM

Consider a room of dimensions $l \times w \times h$ (length, width, and height, respectively). A Cartesian reference frame is connected with the center of this room and axes are made parallel to the walls so that the six walls have coordinates $x = \pm l/2$, $y = \pm w/2$, $z = \pm h/2$. A monopole source of strength Q_0 is placed inside the room at $x = x_0$, $y = y_0$, $z = z_0$. In the absence of walls, the source would generate the acoustic field

$$\psi_0(\mathbf{r}; k) = \frac{Q_0}{4\pi|\mathbf{r} - \mathbf{r}_0|} e^{ik|\mathbf{r} - \mathbf{r}_0|} \quad (1)$$

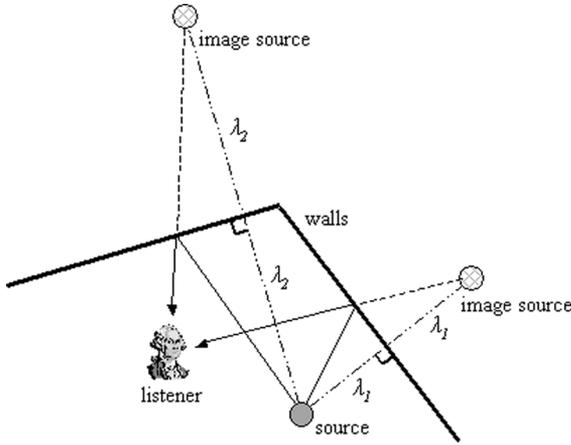


Fig. 1. Illustration of the formation of image sources. For clarity, all illustrative images are shown in two dimensions, though the computations are performed in three dimensions.

where $\mathbf{r}_0 = (x_0, y_0, z_0)$ is a vector of the source position. This field satisfies the Helmholtz equation with Sommerfeld radiation condition

$$\begin{aligned} \nabla^2 \psi_0 + k^2 \psi_0 &= -\delta(|\mathbf{r} - \mathbf{r}_0|) \\ \lim_{r \rightarrow \infty} r \left(\frac{\partial \psi_0}{\partial r} - ik\psi_0 \right) &= 0. \end{aligned} \quad (2)$$

Equation (2) is obtained via the Fourier transform of the wave equation. Here $k = \omega/c$ is the wavenumber, ω is the frequency, and c is the sound speed.

The presence of walls requires the pressure field to satisfy boundary conditions at them. Usually a no flux boundary condition is imposed corresponding to sound hard boundaries,

$$\frac{\partial \psi}{\partial n} = 0. \quad (3)$$

(This boundary condition results in a non-decaying sound field; to avoid such nonphysical behavior, wall absorption was heuristically introduced in [11]. Other boundary conditions are also possible, but are not considered in [11].)

With a monopole source in a rectangular room, the boundary condition (3) results in a specular reflection of sound waves. A reflection of the sound from the wall can equivalently be represented by introduction of a monopole *image source*, which is placed behind the wall along the perpendicular from the actual source to the wall so that the distances from the wall to the actual source and to the image source are equal (Fig. 1). The waves emitted by the image source are also reflected off other walls, creating higher order reflections, which are in turn represented by other image sources. The location of these image sources is denoted as \mathbf{r}_q , $q = 1, 2, \dots$, and there are an infinite number of them. Simple geometric relationships can be employed to find the image source positions. The computations are especially easy for the rectangular-shaped room considered here, in which case a simple, regular, infinite lattice is formed

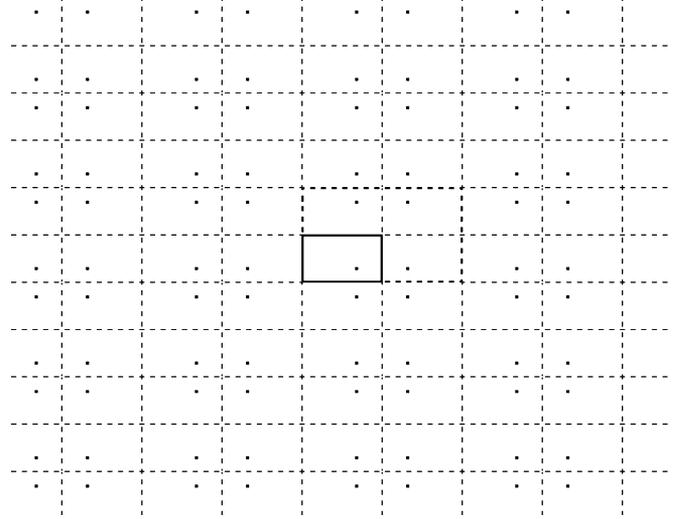


Fig. 2. Illustration to formation of the infinite lattice of image sources in case of a two-dimensional rectangular room. The room is shown as a solid rectangle in the middle, and the dot inside the room is the actual sound source. All other dots are image sources. The lattice is formed by infinitely repeating the block surrounded by thick dashed line.

by the image sources (Fig. 2 shows an illustration, drawn in two dimensions for simplification).

In practice the walls are not ideal but rather have impedance and therefore absorb part of the sound. In [11] wall absorption is simulated by introducing six heuristic frequency-independent real-valued wall reflection coefficients β_i , $0 \leq \beta_i < 1$, $i = 1, \dots, 6$ (one per room wall), and the strength Q_q of the q th image source is obtained by multiplying the strength of the actual source Q_0 (which can be taken to be unit, $Q_0 = 1$) by the reflection coefficients for all walls in which the actual source was reflected to create that q th image source. As discussed in [11], such a heuristic approach corresponds to the assumption of a nonphysical directional absorption model. Despite the criticism, it has been used by several authors to model room acoustics. (Another, better physically-justified approach is to make the reflection coefficient complex and dependent on the incidence angle, which is in fact necessary to correctly simulate the interaction between the plane wave and the surface [4]. This approach is not followed here, but the proposed algorithm can handle the case of angle-dependent reflection coefficients as well).

Because of wall absorption, the image source strength decreases as more and more orders of reflections are considered. Furthermore, in any practical system the reverberation duration must be limited. Let t_{\max} be the maximum time considered. Due to finite propagation speed of acoustic waves, only image sources that lie within $|\mathbf{r}_q - \mathbf{r}| \leq ct_{\max}$ influence the sound field in the room, where \mathbf{r} is the radius vector of an arbitrary point inside the room.

Let $D_{\min} = (1/2)\sqrt{l^2 + w^2 + h^2}$ be the distance from the room center to any room corner (a vertex of the rectangular box). Let S_{\min} be the sphere with radius D_{\min} centered at the origin. This sphere includes the entire room. Consider also a sphere S_{\max} of radius $R_{\max} = ct_{\max} + D_{\min}$ concentric to S_{\min} . Assume that there are total of N_{\max} image sources located inside

S_{\max} . The image sources located outside S_{\max} do not influence the sound field in the room for time interval $[0, t_{\max}]$. Therefore the field inside the room within specified time limits can be represented as

$$\psi(\mathbf{r}; k) = \sum_{|\mathbf{r}_q| < R_{\max}} \psi_q(\mathbf{r}; k), \quad q = 0, \dots, N_{\max} \quad (4)$$

where

$$\psi_q(\mathbf{r}; k) = \frac{Q_q}{4\pi|\mathbf{r} - \mathbf{r}_q|} e^{ik|\mathbf{r} - \mathbf{r}_q|}. \quad (5)$$

The RTF, by definition, is the potential at a point \mathbf{r} generated by a monopole source at a point \mathbf{r}_0 , which is exactly what is expressed by (4). As such, the RTF is provided by (4), and actual RIR can be calculated by taking the inverse Fourier transform

$$s(\mathbf{r}; t) = \sum_{|\mathbf{r}_q| < R_{\max}} \frac{Q_q \delta\left(t - \frac{|\mathbf{r} - \mathbf{r}_q|}{c}\right)}{4\pi|\mathbf{r} - \mathbf{r}_q|}. \quad (6)$$

In fact, this expression accurately describes, up to the scale factor, the process of assembling the time-domain impulse response in [11]. Finally, once the RIR for the source and the receiver positions is computed using (4) or (6), one can convolve the source signal with it and obtain the reverberant signal that would be presented at the receiver in the simulated room.

Since the development of the Allen–Berkley method, much progress was achieved in room reverberation simulation. It is well-known now that more sophisticated methods that are able to handle complex non-rectangular rooms, sound diffraction, diffuse reflections, and grazing incidence effects must be used to simulate sound fields that would agree with the measured RIRs with good accuracy [39]–[41]. Also, statistical ray-tracing methods (e.g., [42]) and beam-tracing algorithms (e.g., [20]) are often employed instead of image-based methods. Still, despite its shortcomings, Allen–Berkley algorithm is simple, can be implemented quickly, compares favorably in terms of cost with commercially available room simulation software, and (most important) produces simulated reverberation that is often perfectly acceptable, depending on the purpose of simulation. Because of that, it is still in broad use and is being widely cited by the research community. In this paper, a reformulation of the algorithm aimed at reducing its computational complexity (especially in case of multiple sources and/or multiple receivers presented in the room) is provided.

III. MULTIPOLE METHOD

Despite its computational complexity growing as the product of the number of sources and the number of receivers, the practical implementation of the time-domain solution for commonly occurring scenarios using (6) is fast. However, it cannot handle the case when wall reflection coefficients are frequency-dependent. To account for possible variations in image source spectra, it is necessary to implement the Allen–Berkley algorithm in the frequency domain by using the (4) for many wavenumbers (in fact, half as many as there are points in RIR) to obtain the RTF

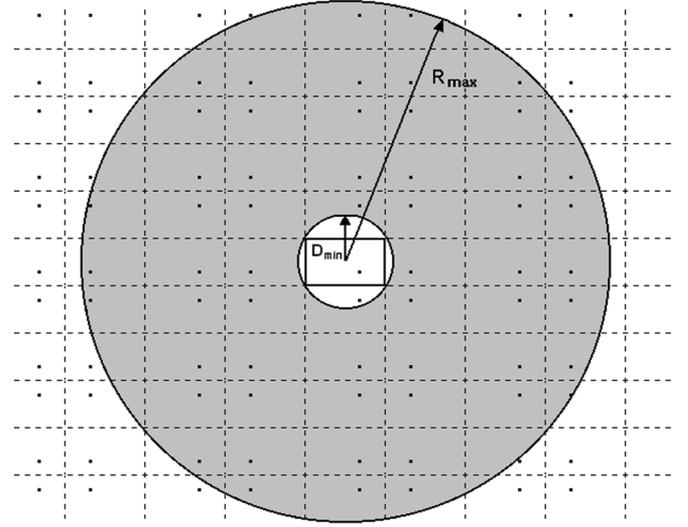


Fig. 3. Part of the infinite lattice of the image sources. A sphere of radius D_{\min} encloses the room. The sphere of radius R_{\max} represents the maximum radius of interest, and image sources farther away are ignored. All the image sources that are located in the grey area between two spheres constitute the regular part of the potential. The actual source in the room and one image source that is located outside of the room but inside the smaller sphere constitute the singular part of the potential.

and then take inverse Fourier transform to get the RIR. In this section, an algorithm to speed up a frequency-domain implementation of the Allen–Berkley method is presented.

The practical shortcoming of the solution (4) is that the number N_{\max} of image sources within a sphere S_{\max} can be large (tens of thousands) and evaluation of the sum over N sources for a given location in the room can be computationally expensive. Furthermore, the RTF often has to be computed at M points in the room, in which case direct implementation of (4) requires that for each of the M evaluation points N sources be examined, resulting in a total of $O(NM)$ operations.

A fast method for evaluation of the sum (4) is proposed and implemented here. The method uses a multipole expansion technique and requires only $O(N + M)$ operations for the sound field evaluation.

First, the sum (4) is broken into two terms

$$\psi(\mathbf{r}) = \psi_s(\mathbf{r}) + \psi_r(\mathbf{r}) \quad (7)$$

(the dependence on k is suppressed from now on for clarity), where ψ_s and ψ_r are the singular and the regular parts, respectively, of the field inside the sphere S_D of radius $D \geq D_{\min}$ concentric to S_{\min} . ψ_s and ψ_r are given by

$$\psi_s(\mathbf{r}) = \sum_{r_q \leq D} \psi_q(\mathbf{r}), \quad \psi_r(\mathbf{r}) = \sum_{D < r_q < R_{\max}} \psi_q(\mathbf{r}). \quad (8)$$

The singular part is generated by the actual source and by the image sources located inside the sphere S_D . The regular part is generated by the image sources located in the domain bounded by spheres S_D and S_{\max} (Fig. 3; in the figure, $D = D_{\min}$, which is shown later to be the optimal choice for D ; however, all derivations here are valid for any $D \geq D_{\min}$). Below the evaluation of the regular part $\psi_r(\mathbf{r})$ is considered.

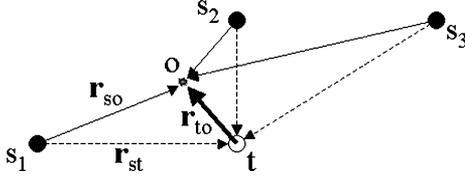


Fig. 4. Illustration of the translation operation. Monopole sources s_1 , s_2 , and s_3 create some potential at the evaluation point o . Instead of evaluating it directly as $\sum \psi(\mathbf{r}_{so})$, the potential of each source is expanded (represented) in terms of the functions centered about another point t (center of expansion); \mathbf{r}_{st} denotes the translation vector and \mathbf{r}_{to} is the vector from the center of expansion to the evaluation point. Expansion at t is done for all three sources, and their expansion coefficients are summed to form the multipole source at t . Then, $\sum \psi(\mathbf{r}_{so})$ can be computed by evaluating the potential of that multipole source at point o . Vectors \mathbf{r}_{so} and \mathbf{r}_{st} for sources 2 and 3 are not labeled for clarity.

Assume that a unit strength monopole source is located at a point s . Its potential $\psi(\mathbf{r}_{so})$ at any point o is given by the formula

$$\psi(\mathbf{r}_{so}) = \frac{e^{ikr_{so}}}{4\pi|\mathbf{r}_{so}|} \quad (9)$$

[compare with (1)] and can be expanded as the product of a series of singular and regular basis functions of the Helmholtz equation centered about another point t using the factorization identity [38]

$$\psi(\mathbf{r}_{so}) = ik \sum_{n=0}^{\infty} \sum_{m=-n}^{m=n} S_n^{-m}(-\mathbf{r}_{st}) R_n^m(\mathbf{r}_{to}), \quad r_{to} \leq r_{st} \quad (10)$$

where \mathbf{r}_{so} denotes the vector from the monopole source location s to the evaluation point o , \mathbf{r}_{st} denotes a *translation vector* from the original monopole source location s to the center of expansion t , \mathbf{r}_{to} denotes a vector from the center of expansion t to the evaluation point o (Fig. 4), and

$$\begin{aligned} R_n^m(\mathbf{r}) &= j_n(kr) Y_n^m(\theta, \varphi), \\ S_n^m(\mathbf{r}) &= h_n(kr) Y_n^m(\theta, \varphi), \\ Y_n^m(\theta, \varphi) &= (-1)^m \sqrt{\frac{2n+1}{4\pi} \frac{(n-|m|)!}{(n+|m|)!}} \\ &\quad \times P_n^{|m|}(\cos \theta) e^{im\varphi}. \end{aligned} \quad (11)$$

Here, (r, θ, φ) are the spherical coordinates of the radius vector \mathbf{r} , $j_n(kr)$ and $h_n(kr)$ are the spherical Bessel and spherical Hankel functions, respectively, $Y_n^m(\theta, \varphi)$ are the orthonormal spherical harmonics, and $P_n^m(\mu)$ are the associated Legendre functions.

Next, all the image sources comprising $\psi_r(\mathbf{r})$ in (8) are expanded at the origin using (10). For each individual image source, the point s in (10) corresponds to its location and the point t —to the origin (center of the room). These expansions are then substituted into the sum for the regular part of the potential field $\psi_r(\mathbf{r})$ in (8)

$$\psi_r(\mathbf{r}) = \sum_{D < r_q < R_{\max}} \left[ik \sum_{n=0}^{\infty} \sum_{m=-n}^{m=n} Q_q S_n^{-m}(\mathbf{r}_q) R_n^m(\mathbf{r}) \right] \quad (12)$$

and it is possible to regroup the double summation to form the coefficients C_n^m as follows:

$$\psi_r(\mathbf{r}) = ik \sum_{n=0}^{\infty} \sum_{m=-n}^{m=n} C_n^m R_n^m(\mathbf{r}) \quad (13)$$

$$C_n^m = \sum_{D < r_q < R_{\max}} Q_q S_n^{-m}(\mathbf{r}_q) \quad (14)$$

where \mathbf{r} is now just a vector from the origin to the evaluation point and \mathbf{r}_q is a vector from the origin to the position of q th image source. The coefficients C_n^m do not depend on the evaluation point \mathbf{r} and can be precomputed for given image source locations (which are determined in the same manner as in the original Allen–Berkley algorithm—i.e., by simple geometric computations illustrated by Figs. 1 and 2 in the same manner as in the original Allen–Berkley algorithm). In fact, the summation (14) can be viewed as the process of collecting all the image sources constituting the regular part of the potential in (8) into a single (but rather complex) multipole source located at the center of the room. At any point in the room, the potential generated by this multipole source would match exactly the sum of the potentials generated by all collected image sources. As such, $\psi_r(\mathbf{r})$ can now be evaluated as the potential of this multipole source using (13), which does not involve summation over all image sources anymore. As there exist tens of thousands image sources even in the simulation of as short as 100 ms RIR, a very significant speed-up in computation of $\psi_r(\mathbf{r})$ can be achieved.

Equation (13) is exact. These series converge very quickly, and the outer summation can be truncated at $p+1$ terms

$$\psi_r(\mathbf{r}) = ik \sum_{n=0}^p \sum_{m=-n}^{m=n} C_n^m R_n^m(\mathbf{r}). \quad (15)$$

The parameter p is known as the *truncation number*. Evaluation of $\psi_r(\mathbf{r})$ at any given point \mathbf{r} thereby requires a summation of $(p+1)^2$ terms. This must be compared with the summation over N image sources with $D < r_q < R_{\max}$ if (8) were used directly. Note that p depends only on the desired accuracy of the summation, and $(p+1)^2$ can be orders of magnitude smaller than N . Of course, it is still necessary to compute C_n^m first, which does require processing of N image sources; however, if $\psi_r(\mathbf{r})$ is to be computed at M evaluation points, the one-time cost of computing C_n^m is amortized over them, resulting in overall complexity decrease.

As the coefficients C_n^m do not depend on \mathbf{r} , the practical RIR computation at M points is done in two steps. In the first step, C_n^m are computed using (14). In the second step, $\psi_r(\mathbf{r}; k)$ at M points is computed using the set of C_n^m coefficients produced in the first step, and $\psi_s(\mathbf{r}; k)$ is computed directly ($\psi_s(\mathbf{r}; k)$ is generated by the actual source and by a few image sources inside S_D , so the computation time for $\psi_s(\mathbf{r}; k)$ is negligible). $\psi_r(\mathbf{r}; k)$ and $\psi_s(\mathbf{r}; k)$ are then summed up to produce $\psi(\mathbf{r}; k)$, and the final RIR is obtained by inverse Fourier transform.

IV. ERROR EVALUATION

Direct numerical implementation of (15) is complicated by the fact that for large n and fixed \mathbf{r}_q the magnitude of $S_n^{-m}(\mathbf{r}_q)$

grows as n^n . This imposes limitations on the truncation number. For large n and fixed \mathbf{r} and \mathbf{r}_q the principal terms of asymptotic expansions of $S_n^{-m}(\mathbf{r}_q)$ and $R_n^m(\mathbf{r})$ in (11) can be found using asymptotic expressions for Bessel functions from Abramowitz and Stegun [43]

$$\begin{aligned} S_n^{-m}(\mathbf{r}_q) &\sim -\frac{i}{kr_q} \sqrt{\frac{2}{e}} \left(\frac{2n+1}{ekr_q}\right)^n Y_n^{-m}(\theta_q, \varphi_q), \\ R_n^m(\mathbf{r}) &\sim \frac{1}{2n+1} \sqrt{\frac{e}{2}} \left(\frac{ekr}{2n+1}\right)^n Y_n^m(\theta, \varphi). \end{aligned} \quad (16)$$

These expressions allow one to determine the order of truncation [38]. Define

$$p_k(r) = \frac{ekr - 1}{2}. \quad (17)$$

As can be seen from (16), for given r_q and $n > p_k(r_q)$ functions $S_n^{-m}(\mathbf{r}_q)$ grow exponentially. At the same time the product $S_n^{-m}(\mathbf{r}_q)R_n^m(\mathbf{r})$ is finite and for any $r/r_q < 1$ converges fast to zero as n increases. It can be evaluated for large n and fixed \mathbf{r} and \mathbf{r}_q using (16)

$$S_n^{-m}(\mathbf{r}_q)R_n^m(\mathbf{r}) = \left(\frac{r}{r_q}\right)^n \frac{Y_n^{-m}(\theta_q, \varphi_q)Y_n^m(\theta, \varphi)}{ikr_q(2n+1)}. \quad (18)$$

This evaluation shows that the truncation error decreases with n geometrically. It also depends on the ratio r/r_q , which decays for sources located far from the evaluation points. The maximum value of r/r_q is realized for sources at $r_q \sim D$. Therefore, the optimal selection of truncation number as

$$p = \mu p_k(D) \quad (19)$$

$\mu = 1$, provides both stable computation of the series for the entire domain and the following evaluation of the truncation error for any point inside the room

$$\epsilon_p = O\left(\frac{1}{(2p+1)^2} \left(\frac{D_{\min}}{D}\right)^p\right). \quad (20)$$

It also follows that to minimize p one should set $D = D_{\min}$.

The obtained result for the optimal value of the truncation number p is roughly consistent with the estimations presented earlier in the literature. For example, in [44] rough lower and upper bounds are established on the truncation number as follows: first, the truncation number should be larger than $2kD$ for the multipole expansion to be valid (i.e., to produce the correct results) in the region of radius D ; and second, the truncation number should not be much larger than $\min kr_{st}$, where r_{st} is the length of the translation vector \mathbf{r}_{st} in (10) (in other words, if the nearest image source is located at distance r^* from the origin and thus the length of the shortest translation is r^* , then p should not be much larger than kr^* ; otherwise, large-magnitude oscillations in the spherical Hankel functions cause loss of accuracy). Two important conclusions follow from these bounds:

- If the nearest image source is located at the distance which is comparable to the room radius D_{\min} (which is usually true, as can be observed from Fig. 3), the upper and lower

bounds derived in [44] roughly coincide and approximately agree with (19).

- If D_{\min} is reduced, p can also be reduced.

The second conclusion is important for the room subdivision technique described in Section VI.

The error bounds shown here suggest the selection of the truncation number $p_k(r) \simeq (e/2)kr$ [given by (17)]. They are sufficiently accurate and are verified experimentally in Section VII. It should be noted that more accurate error bounds could be derived; they are not used here to avoid complicating the presentation unnecessarily [see, e.g., [38] where it is shown that for the desired truncation error ϵ and large kr it is sufficient to select $p_k(r) = kr + \eta_\epsilon(kr)^{1/3}$, which is asymptotically smaller than $(e/2)kr$ (η_ϵ is an ϵ -dependent constant)].

V. COMPUTATIONAL COMPLEXITY

In this section, the computational complexity of the Allen–Berkley algorithm and of the proposed multipole algorithm is considered. The complexity is first expressed in terms of the number of sources N and the number of evaluation points M . In the Allen–Berkley algorithm, computation of the field potential calls for repeated evaluation of the potential for each source–receiver pair, resulting in $O(NM)$ operations. In contrast, when the field potential is computed using the multipole algorithm, the first algorithm step—the computation of C_n^m —takes $O(N)$ time (because the sum (14) is taken over N image sources) and the second step takes $O(M)$ time (because the truncated series (15) is computed once for each receiver point) for a total of $O(N + M)$ operations. The complexity decrease is more significant when M is large. The constant in the big- O notation depends on the truncation number p (i.e., on the desired accuracy).

The regular structure of the image source lattice also permits evaluation of the computational complexity in terms of the simulated reverberation length t_{\max} . Note first that R_{\max} is roughly proportional to t_{\max} and that the number of image sources in the S_{\max} grows as R_{\max}^3 (because the problem is three-dimensional). The time-domain Allen–Berkley algorithm performs a summation of impulses generated by all image sources, and therefore its running time for M evaluation points is simply $O(Mt_{\max}^3)$.

The Allen–Berkley algorithm can also be implemented straightforwardly in the frequency domain by computing the field potential for all wavenumbers k and then taking the inverse Fourier transform to obtain RIR. Computation for one k requires $O(t_{\max}^3)$ time (per receiver point); the number of distinct k is proportional to the RIR length, and therefore the total running time (per receiver point) of the frequency-domain Allen–Berkley algorithm is $O(t_{\max}^4)$. This running time is denoted T_{AB} . If the field is to be evaluated at M receivers, the total running time is $MT_{AB} = O(Mt_{\max}^4)$.

A faster hybrid-domain implementation of Allen–Berkley algorithm is possible. It is based on computing, in the frequency domain, the spectrum for each image source using a limited number of frequency bands (e.g., 128 bands), obtaining the 256-point IR for that image source by IFFT, and inserting computed IRs for all image sources into the final RIR with appropriate time delays (determined by the range of each image source) [45].

Because the number of the frequency bands is fixed, the total computation time grows only as $T'_{AB} = O(t_{\max}^3)$. However, it should be noted here that the constant hidden in the big-O notation for T'_{AB} is quite large because the computation is done at a fixed but pretty large (e.g., 128) number of frequencies and because the 256-point IFFT must be done for each image source. In the straightforward frequency-domain implementation the length of IFFT is the same as the length of RIR, but that IFFT is done only once after the contributions of all image sources are assembled in the frequency domain. It will be shown in Section VIII that because of the reasons mentioned, the difference between T_{AB} and T'_{AB} is not as dramatic as it can be expected from $O(t_{\max}^3)$ and $O(t_{\max}^4)$ bounds. Similar technique with an IIR (instead of a FIR) filter is used in [21].

In contrast, the running time of the proposed multipole algorithm consists of two parts, T_{MP1} and T_{MP2} . The first part T_{MP1} (the multipole source setup time) represents the computational time necessary to translate all the image sources to the origin using (14); this procedure is done only once, no matter how many evaluation points there are in the room. The second part T_{MP2} represents the time necessary to compute the RIR for one evaluation point given that coefficients C_n^m are already available.

Consider first the complexity for the case where the offline RIR computation must be performed at multiple locations. If there are M such locations, the total running time is $T_{MP1} + MT_{MP2}$. As the first step of the multipole algorithm goes over all sources in S_{\max} for all values of k , T_{MP1} is $O(t_{\max}^4)$. The second step however no longer involves processing all the image sources but rather computes the value of the multipole expansion for all k so that T_{MP2} is $O(t_{\max})$. The total running time of the multipole algorithm is then $O(t_{\max}^4 + Mt_{\max})$. It is evident that when M is sufficiently large, the multipole algorithm is faster than the frequency-domain and the time-domain Allen–Berkley algorithms.

In the second situation the evaluation of the RTF is performed online at a receiver location that is changing in real-time (e.g., for MFA beamforming). In this situation it is possible to neglect the running time of the first step of the multipole algorithm (for example, when the set of C_n^m for a given room and a given receiver (or source) position is computed in advance and all that matters is the latency of RTF update in response to the source (or receiver, respectively) motion), and the running time (per evaluation point) of the time-domain (frequency-domain, hybrid-domain) Allen–Berkley algorithm grows as t_{\max}^3 (t_{\max}^4 , t_{\max}^3 , respectively), whereas the running time (per evaluation point) of the multipole method grows only as t_{\max} .

However, the performance evaluation results (presented in Section VIII) reveal that for reasonable values of t_{\max} and M the multipole algorithm is significantly slower than the time-domain Allen–Berkley algorithm because of the computations of the special functions involved in the multipole expansion (10). To address this deficiency, a room subdivision method for improving the multipole algorithm performance is developed and is described in the next section. It should also be noted that if the simulation task necessitates the use of the frequency-domain algorithm, even the straightforward implementation of the proposed multipole algorithm is feasible and has an advantage

in speed (also demonstrated in Section VIII) over the hybrid-domain Allen–Berkley algorithm for commonly encountered values of t_{\max} and M .

VI. ROOM SUBDIVISION TECHNIQUE

Due to computational precision issues, it is problematic to raise the truncation number beyond a certain value on a typical computer. It also may be desirable to speed up RIR computation to compete successfully with the time-domain Allen–Berkley algorithm. To decrease the required truncation number and to decrease T_{MP2} (at the expense of increasing T_{MP1}), a room subdivision technique can be applied. The idea of the technique is to split the room into several subrooms and to create, in each subroom, a separate multipole source that is still constructed from *all* image sources but is used to evaluate the potential only for the evaluation points that belong to that subroom. Because the subrooms are smaller, the radius of validity of these multipole sources can be smaller. In this way, D_{\min} and therefore p can be decreased.

Assume that each side of the room is divided into K pieces, creating a total of K^3 subrooms. Because subroom boundaries are imaginary, the image source lattice formation process (Fig. 3) does not change. Then, for each subroom, all the image sources comprising the regular part of the potential are expanded in terms of the multipole source located at the subroom center, creating K^3 multipole sources, one per subroom. Each multipole source is now a local representation of all image sources (by “local”, it is meant that the representation is valid only in the corresponding subroom). When a RTF is to be evaluated at some point, the subroom to which that point belongs is first found, and then the RTF is obtained by computing the potential of that subroom multipole source at the evaluation point and adding the singular part of the potential computed directly as before. In other words, each subroom created by subdivision is treated by the algorithm exactly in the same manner as the original room is treated in no subdivision case.

There are two advantages in using the room subdivision technique. Firstly, when the room is divided into K^3 subrooms, D_{\min} gets divided by K . Therefore p also gets divided by K . This allows for computing the RTF at higher frequencies without hitting the machine precision limits (described in Section VIII). Secondly, the number of terms in the multipole expansion is proportional to p^2 , and therefore T_{MP2} , as well as the time to compute C_n^m for one subroom, gets divided by K^2 , allowing for faster RTF update in response to the source/receiver motion in online applications. Because there are total of K^3 subrooms, the T_{MP1} increases by a factor of K . However, this part of the calculation can be performed in advance. The room subdivision technique can therefore greatly increase the algorithm performance in situations where the sets of C_n^m can be precomputed beforehand so that T_{MP1} can be ignored and only T_{MP2} is of importance, and with appropriate subdivision the multipole algorithm can produce RTF faster than not only frequency-domain but also time-domain Allen–Berkley algorithm. Also, if there is reason not to compute responses in some parts of the room, then the computations for the corresponding subrooms can be avoided.

Room subdivision is also useful if the task is offline (so that T_{MP1} cannot be ignored) and the number of evaluation points is large. Consider the case of computing the RIR at M points (e.g., for offline testing of the source localization or speech enhancement algorithm) so that the total computation time is $T_{MP} = T_{MP1} + MT_{MP2}$. As the first term increases and the second term decreases as K increases, it is possible to minimize T_{MP} by choosing K appropriately. Note that the first term grows proportionally to K and the second term grows proportionally to K^{-2} . Assume that the total number of sources is N ; then, T_{MP} can be written as $NK + MK^{-2}$ (up to the scale factor). T_{MP} is minimized when its derivative with respect to K is equal to zero, and that derivative is $N - 2MK^{-3}$. Then, the optimal $K = (2M/N)^{1/3}$. It follows that for offline applications room subdivision is justified only when the number of evaluation points is comparable to the number of sources, which is usually not the case for RIR computation due to the large number of image sources.

VII. RESULTS

To check the accuracy of the proposed algorithm, an implementation in the C programming language was developed and tested against the time-domain and against the frequency-domain implementations of the Allen–Berkley algorithm. Some representative results are shown below.

The first comparison is performed between the values of the potential obtained at a particular frequency using the exact equation (4) and using the approximation series (15), and the error is analyzed for various frequencies and for various truncation numbers p . The computations are carried out for the room ($l = 2.5$ m, $w = 2.5$ m, $h = 2.0$ m, $D_{\min} = 2.03$ m) with the reflection coefficients equal to 0.9 for room walls and 0.7 for the floor and the ceiling. The monopole source is placed at a random position inside the room. The radius $R_{\max} = 40$ m corresponds to $t_{\max} \approx 100$ ms. In this particular configuration, approximately 21500 image sources are located within the sphere of radius R_{\max} .

To understand the behavior of the approximation error, M evaluation points are then distributed randomly over the room, and the field potential at these points is evaluated for a particular frequency and for varying truncation numbers, first by applying frequency-domain Allen–Berkley algorithm summarized by (4) and (5) (i.e., by summing Green's functions of the actual source and of all the image sources) and then by applying the developed algorithm. The difference between the field potential values obtained by two algorithms is the approximation error. Its root mean square approximation error is shown in Fig. 5 for eight different frequencies and for the truncation number varying from 5 to 150. The Fig. 5 legend also provides the optimal truncation number (computed using (19) with $\mu = 1$ and $D = D_{\min}$) for each frequency. It can be seen from the plot that increasing μ beyond 1 does not have detrimental effect on computation accuracy (however, it does slow down computations significantly as more coefficients are to be computed). It can also be seen that the error is indeed negligible for $\mu = 1$, and an approximation of reasonable quality can be obtained for lower μ . For example, the average approximation error is roughly 10%, 1%, 0.1%, and

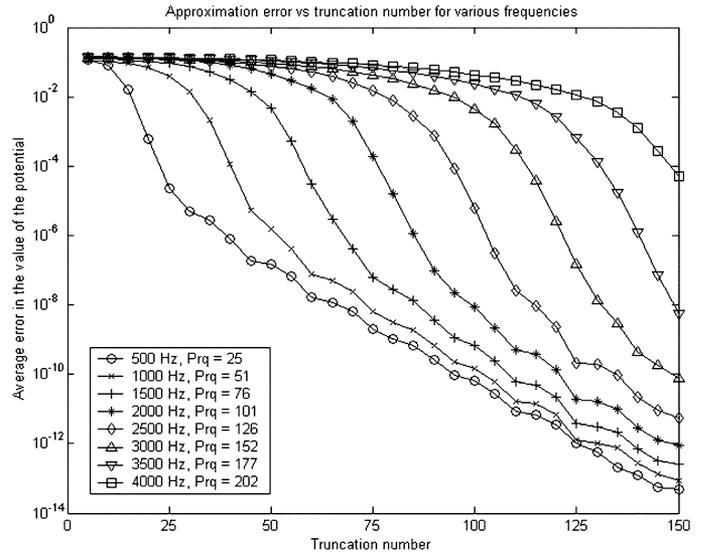


Fig. 5. Approximation error versus truncation number for various frequencies. For each frequency, the optimal truncation number Prq computed using (19) with $\mu = 1$ and $D = D_{\min}$ is shown in the legend.

0.01% for $\mu = 1/2, 2/3, 3/4$, and $4/5$, respectively. As it is expensive to compute multipole expansion coefficients of high orders, in practical applications the compromise can be achieved by selecting the value of $\mu \leq 1$ depending on the desired accuracy, computational power available, and machine precision.

The second comparison is done between the actual RIRs obtained with the traditional time-domain Allen–Berkley method, frequency-domain Allen–Berkley method, and the proposed multipole method. The method for computing the RIR in the frequency domain is as follows. Assume that a RIR of length N_d points sampled at the frequency f_d is desired. (In practice, the radius R_{\max} should then be not more than cN_d/f_d without the correction factor of D_{\min} ; otherwise, image sources that have no effect on the RIR would be unnecessarily generated and included in computations, and the RIR can be significantly contaminated by them because of the wrap-around continuity of the Fourier transform). The RTF then is simply a discrete Fourier transform (DFT) of RIR and has a total of $N_d/2 + 1$ complex components corresponding to the $N_d/2 + 1$ discrete frequencies $0, f_d/N_d, 2f_d/N_d, \dots, f_d/2$. For each of these, excluding the zero frequency, the complex value of the potential at the evaluation point is computed using either (4) or (5), and then the inverse discrete Fourier transform (IDFT) is applied to generate the RIR.

Fig. 6 shows a comparison between RIRs produced by three algorithms at two randomly selected evaluation points. The discretization frequency here is 8 kHz, D is equal to D_{\min} , and μ is set to $3/4$ based on the error analysis presented above. For all three algorithms, time-domain highpass filtering (as described in [11]) with a normalized cut-off frequency of 0.01 is applied to compensate for the nonphysical response at low frequencies due to no-flux boundary conditions. (Obviously, in practice, for frequency-domain methods efficient filtering can be performed directly before doing the inverse Fourier transform; this is not done here for fairness of comparison). For the first pair of source–receiver positions, RIR plots are shown in

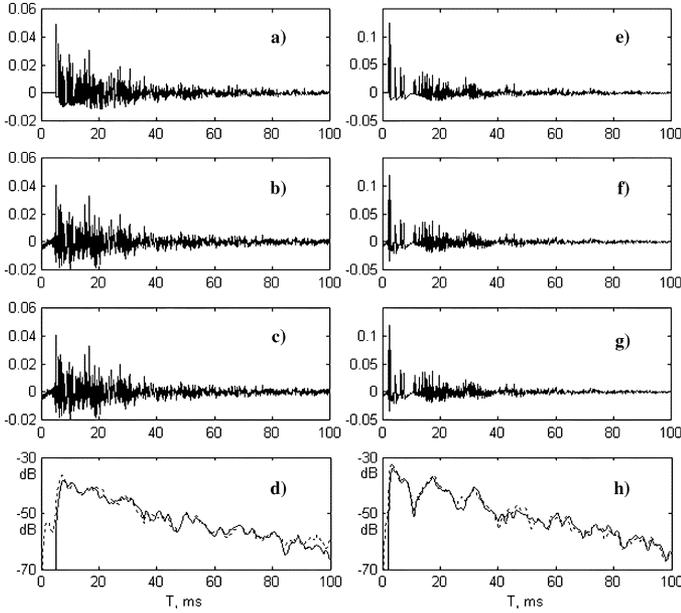


Fig. 6. (a) RIR computed using time-domain Allen–Berkley algorithm for some source–receiver position pair. (b) RIR computed using frequency-domain Allen–Berkley algorithm for the same source–receiver position pair. (c) RIR computed using the proposed multipole algorithm for the same source and receiver positions. (d) Magnitude of the temporal envelope for RIR (a) (solid line) and for RIRs (b) and (c) (dashed line). (e)–(h) Same as (a)–(d) for another pair of source and receiver positions.

the left column of Fig. 6. It can be seen that there is not much difference in the results of the three algorithms, and subplots (b) and (c) are visually identical. Indeed, examination of the actual values of RIR samples computed by the frequency-domain Allen–Berkley algorithm and by the proposed multipole algorithm reveals that they are generally within 0.1% of each other, consistent with the error analysis for $\mu = 3/4$. Subplot (d) shows the temporal envelope of the RIR on log-magnitude scale. The temporal envelope is extracted here by filtering the absolute value of RIR with the second-order lowpass Butterworth filter with a normalized cut-off frequency of 0.025 (i.e., 200 Hz). The solid line in subplot (d) shows the envelope of the RIR produced by the time-domain Allen–Berkley algorithm, and the dashed line represents the envelopes of the RIRs produced by the frequency-domain Allen–Berkley algorithm and by the proposed multipole algorithm, which were found to be identical for plotting purposes at the given scale. The conclusions derived from subplots (a)–(d) also hold for the plots corresponding to the second pair of source–receiver positions (shown in the right column of Fig. 6).

Some differences in peak heights can be seen between the outputs of the time-domain and the frequency-domain Allen–Berkley algorithms. This is due to the fact that the time-domain algorithm rounds the echo arrival time to the nearest sample and places a single peak at that sample, whereas the frequency-domain algorithm is not obligated to do that, distributing the echo over several samples as necessary to preserve its exact arrival time. This effect was pointed out and handled by assignment of the low-passed impulse (instead of just the shifted impulse) to each individual echo in [13]; in

the frequency-domain algorithm, this problem is taken care of automatically. There is also a nonphysical artifact in the outputs of the frequency-domain and of the multipole algorithms—the RIR is not zero before the direct sound arrival due to the RIR tail wrapping around, which is most obvious on RIR envelope plots. The wrap-around influence can be reduced to an arbitrary low level by computing longer and/or zero-padded RIR by appropriately choosing N_d and R_{\max} . Of these two, N_d determines the length of IDFT and R_{\max} determines how long the actual RIR is before IDFT wrap-around (if any). These parameters can be set independently with understanding of consequences (e.g., if R_{\max} is more than cN_d/f_d , then the actual RIR is longer than N_d and the IDFT output is the RIR cut to the length N_d and contaminated by the wrapped-around RIR tail, and if R_{\max} is half of cN_d/f_d , then the length of the actual RIR is approximately $N_d/2$ and the IDFT output is the actual RIR zero-padded to length N_d). However, it is necessary to increase N_d to compute longer and/or zero-padded RIR, which requires finer frequency resolution and correspondingly higher computational load. An alternative method that does not require finer-resolution RTF computation is to simply force the RIR to be zero before the direct arrival time (computed from the source and the receiver positions), which will remove the most obvious and most problematic manifestation of the wrap-around effect (the non-causal artifact at the start of the RIR). The remaining contamination of the rest of the RIR by the wrapped-around RIR tail is likely to be insignificant due to low energy in the wrapped-around part of the response.

VIII. PERFORMANCE AND IMPLEMENTATION CONSIDERATIONS

As computational speed was the primary reason for the development of the proposed algorithm, the running time measurements are done in several simulation runs for variable modeled reverberation length. The simulated room was not changed from the setup described in the previous section. The discretization frequency of 8 kHz, D equal to D_{\min} , and μ equal to $3/4$ also did not change. Thus, for the maximum simulation frequency of 4 kHz $p = 151$, which is close to the maximum possible truncation number if computations are to be done on an Intel CPU based computer.

The running time of the frequency-domain Allen–Berkley algorithm, which essentially performs summation of the Green’s functions of individual image sources, is linear (per one receiver point) in the number of the image sources. In contrast, the running time of the multipole algorithm consists of a fixed part T_{MP1} that is linear in the number of the image sources and variable part MT_{MP2} that is linear in the number of receivers and independent of the number of image sources. If $T_{MP2} < T_{AB}$, then the multipole algorithm has a distinct advantage over the summation of Green’s functions because it becomes possible to compute the RTF faster per evaluation point (which is definitely useful in a real-time application such as MFA beamforming), albeit perhaps at an expense of a significant pre-computation time T_{MP1} . Furthermore, even in the case of offline computations, it might be necessary to compute the RTF at M points, and if $T_{MP2} < T_{AB}$, then at some M_0 (the break-even point) the total computation time of the multipole algorithm $T_{MP1} + M_0T_{MP2}$

TABLE I

RUNNING TIME (PER EVALUATION POINT) FOR THE FREQUENCY-DOMAIN ALLEN-BERKLEY (AB) ALGORITHM (T_{AB}), HYBRID-DOMAIN AB ALGORITHM (T'_{AB}), AND THE PROPOSED MULTIPOLE ALGORITHM (T_{MP2}) FOR VARIABLE RADIUS OF INTEREST R_{\max} AND REVERBERATION LENGTH t_{\max} . T_{MP1} IS THE TIME SPENT BY THE MULTIPOLE ALGORITHM IN THE PRE-CALCULATION PHASE, WHICH IS INDEPENDENT OF THE EVALUATION POINT LOCATION. THE TOTAL RUNNING TIME OF THE MULTIPOLE ALGORITHM IS $T_{MP1} + MT_{MP2}$, WHERE M IS THE NUMBER OF EVALUATION POINTS

R_{\max}, m	t_{\max}, ms	N	T_{AB}, s	T'_{AB}, s	T_{MP1}, s	T_{MP2}, s	M_0
20	~50	2680	0.37	0.25	1775	0.59	N/A
30	~75	9054	1.90	0.89	8551	0.90	8551
40	~100	21438	5.98	2.03	26714	1.21	5600
50	~125	41910	14.36	3.78	64577	1.46	5005
60	~150	72368	30.68	6.34	137100	1.79	4746

will be equal to $M_0 T_{AB}$, and for any $M > M_0$ it will be advantageous to use the multipole algorithm despite the lengthy first step.

Table I shows some experimental results for variable reverberation length. It can be seen from the table that the multipole algorithm is slower per point for 50 ms reverberation, but becomes faster per point for longer reverberation time, with the break-even point M_0 steady decreasing as the reverberation length increases. It can also be noted that, as predicted in Section V, T_{AB} grows as t_{\max}^4 whereas T_{MP2} grows only as t_{\max} . The multipole setup time T_{MP1} grows at the same rate as T_{AB} .

For reference, the running time T'_{AB} of the hybrid-domain Allen-Berkley algorithm is also presented in the Table I. The number of frequency bands for the hybrid-domain algorithm is equal to 32 (so that for each image source its 64-point IR is computed). The hybrid implementation is clearly much faster than the straightforward frequency-domain implementation; however, T_{MP2} still grows slower than T'_{AB} , which can be expected from big-O bounds.

Still, the running time of the multipole algorithm is large in comparison with the time-domain Allen-Berkley algorithm (for example, it takes only 16.0 ms to compute the room RIR using time-domain Allen-Berkley algorithm for $R_{\max} = 60$ m) because the time-domain algorithm has an additional advantage of not repeating computations for many frequencies (at a disadvantage of being unable to account for frequency dependencies in sound interaction with surfaces and materials). It should be noted that the proposed multipole algorithm accounts for the contribution of all image sources in only 3.0 ms per one frequency for $R_{\max} = 60$ m, which compares favorably with 16.0 ms required to sum up all the image sources using the Allen-Berkley algorithm. It can also be noted that the values of R_{\max} for which the straightforward implementation (without use of the room subdivision technique) of the multipole algorithm becomes faster per evaluation point than the time-domain Allen-Berkley algorithm are not practical, and T_{MP1} for such large R_{\max} would be extremely large.

The running times given here are obtained with Microsoft Visual C++ 6.0 optimizing compiler with full optimization on a dual-CPU Intel Pentium Xeon 1.7-GHz computer. Computations of the spherical Bessel/Hankel functions and spherical harmonics are based heavily on the routines described in the book [46] and freely available on the World Wide Web.¹ Implementation of the algorithm for high truncation numbers be-

comes problematic because of the machine precision issues and limited range of the numbers that can be stored in double-precision variables on Intel architecture computer. For example, the value of the spherical harmonic $Y_n^m(\theta, \varphi)$ given by (11) is a product of an associated Legendre polynomial (which has very large value for large m and n) and the square root in (11) (which has very small value for large m and n). To avoid expensive constant recomputation of the factorials of large numbers, the value of the square root is precomputed for all possible combinations of m and n up to the maximum possible truncation number using arbitrary precision arithmetic package (to prevent overflow in the computation of the factorial). However, the minimum number that can be stored in a double precision variable on an Intel architecture computer is approximately $2.2 \cdot 10^{-308}$ (all smaller values are represented as zero), and the value of the square root in (11) underflows anyway at $m = n = 155$. As a result, the computed value of some spherical harmonics (specifically, the ones with high m and n) becomes zero if $p > 155$, even given that the value of the associated Legendre polynomial is about 10^{302} for $m = n = 155$ and after multiplication the result has close to unit magnitude. The values of the associated Legendre polynomial and of the imaginary part of the spherical Hankel functions (computed from the values and from the derivatives of the spherical Bessel functions) would also soon present overflow problem if p were to be further increased. Because of these issues, the maximum possible truncation number for the straightforward implementation of the multipole algorithm is 155, with the quality of approximation degrading beyond that point due to zeroing out and/or overflowing of more and more of the basis functions in the multipole expansion, and with $p = 155$ reasonable-quality approximation is obtained for the frequencies up to 6 kHz. Computations with higher p can be carried out by implementing the algorithm with the help of an arbitrary precision arithmetic software package (which is likely to be extremely slow), on a different architecture computer that has wider range of double precision numbers (such as one based on the Motorola 680X0 or Sun SPARC processors, which have range of double precision numbers from 10^{-4932} to 10^{4932}), or even on a specialized hardware.

An alternative method of raising the high frequency limit is to use the room subdivision technique described in Section VI, which allows p to be decreased. Preliminary tests using the simulated setup described above with $K = 2$ and $R_{\max} = 40$ m show that the results obtained are of the same quality as in case of $K = 1$ (i.e., no subdivision) and $p = 150$. Another ad-

¹[Online]. Available: <http://jin.ece.uiuc.edu/routines/routines.html>

vantage of the room subdivision technique is that T_{MP2} gets decreased (at the expense of increasing T_{MP1}). Experiments done with $K = 2$ show that T_{MP1} indeed grows from 26514 to 51223 s (approximately by a factor of K) and T_{MP2} falls from 1.21 to 0.29 s (approximately by a factor of K^2), confirming the theoretical predictions. Note that T_{MP2} now allows for near real-time operations. Additional subdivisions decrease T_{MP2} even further.

IX. CONCLUSION

An algorithm for fast frequency-domain computation of the room transfer function is proposed and tested. The algorithm is based on multipole expansion of the potentials generated by image sources. A comparison with the classical Allen–Berkley algorithm shows that the computations with similar accuracy can be performed at substantial speed-ups.

Further, it is possible to devise a generic “physically-valid” room auralization algorithm based on the numerical solution of the acoustic wave propagation equation in arbitrary-shaped room. Such algorithms exist but have rather limited applicability because of the prohibitively high computational cost, and currently less accurate perception-oriented approximate modeling techniques are usually employed in auralization. Incorporation of the multipole techniques similar to the one described in this paper into physically-valid room auralization algorithms has a potential of significantly reducing associated computational costs and enabling fast and accurate numerical simulation of reverberant acoustic fields.

ACKNOWLEDGMENT

The authors thank the three anonymous reviewers for their helpful and careful comments and suggestions.

REFERENCES

- [1] R. Duraiswami, N. A. Gumerov, D. N. Zotkin, and L. S. Davis, “Efficient evaluation of reverberant sound fields,” in *Proc. IEEE WASPAA*, New Paltz, NY, Oct. 2001, pp. 203–206.
- [2] W. C. Sabine, “Reverberation,” in *Acoustics: Historical and Philosophical Developments*. Stroudsburg, PA: Dowden, Hutchinson, and Ross, 1972, paper published originally in 1900.
- [3] Y. Ando, *Concert Hall Acoustics*. Berlin, Germany: Springer-Verlag, 1985.
- [4] H. Kuttruff, *Room Acoustics*, 3rd ed. Amsterdam, The Netherlands: Elsevier Applied Science, 1991.
- [5] W. M. Hartmann, “Listening in a room and the precedence effect,” in *Binaural and Spatial Hearing in Real and Virtual Environments*, R. H. Gilkey and T. R. Anderson, Eds. Mahwah, NJ: Lawrence Erlbaum, 1997, pp. 191–210.
- [6] B. G. Shinn-Cunningham, J. G. Desloge, and N. Kopco, “Empirical and modeled acoustic transfer functions in a simple room: effects of distance and direction,” in *Proc. IEEE WASPAA*, New Paltz, NY, 2001, pp. 419–423.
- [7] J. L. Flanagan, A. C. Surendran, and E. E. Jan, “Spatially selective sound capture for speech and audio processing,” *Speech Commun.*, vol. 13, no. 1–2, pp. 207–222, 1993.
- [8] J. H. DiBiase, H. F. Silverman, and M. S. Brandstein, “Robust localization in reverberant rooms,” in *Microphone Arrays: Signal Processing Techniques and Applications*, M. S. Brandstein and D. B. Ward, Eds. Berlin, Germany: Springer-Verlag, 2001, pp. 157–180.
- [9] M. R. Schroeder, “Natural sounding artificial reverberation,” *J. Audio Eng. Soc.*, vol. 10, no. 3, pp. 219–223, 1962.
- [10] J. A. Moorer, “About this reverberation business,” *Comput. Music J.*, vol. 3, no. 2, pp. 13–28, 1979.
- [11] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” *J. Acoust. Soc. Amer.*, vol. 65, no. 5, pp. 943–950, 1979.
- [12] J. Borish, “Extension of the image model to arbitrary polyhedra,” *J. Acoust. Soc. Amer.*, vol. 75, no. 6, pp. 1827–1836, 1984.
- [13] P. M. Peterson, “Simulating the response of multiple microphones to a single acoustic source in a reverberant room,” *J. Acoust. Soc. Amer.*, vol. 80, no. 5, pp. 1527–1529, 1986.
- [14] M. Kompis and H. Dillier, “Simulating transfer functions in a reverberant room including source directivity and head-shadow effects,” *J. Acoust. Soc. Amer.*, vol. 93, no. 5, pp. 2779–2787, 1993.
- [15] J. S. Suh and P. A. Nelson, “Measurement of transient response of rooms and comparison with geometrical acoustic models,” *J. Acoust. Soc. Amer.*, vol. 105, no. 4, pp. 2304–2317, 1999.
- [16] T. D. Abhayapala, R. A. Kennedy, and R. C. Williamson, “Noise modeling for nearfield array optimization,” *IEEE Signal Process. Lett.*, vol. 6, no. 8, pp. 210–212, Aug. 1999.
- [17] M. S. Brandstein, “Time-delay estimation of reverberated speech exploiting harmonic structure,” *J. Acoust. Soc. Amer.*, vol. 105, no. 5, pp. 2914–2919, 1999.
- [18] M. Omologo, P. Svaizer, and M. Matassoni, “Environmental conditions and acoustic transduction in hands-free speech recognition,” *Speech Commun.*, vol. 25, no. 1, pp. 75–95, 1998.
- [19] P. W. Shields and D. R. Campbell, “Intelligibility improvements obtained by enhancement method applied to speech corrupted by noise and reverberation,” *Speech Commun.*, vol. 25, no. 1, pp. 165–175, 1998.
- [20] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, M. Sondhi, J. E. West, G. Pingali, P. Min, and A. Ngan, “A beam tracing method for interactive architectural acoustics,” *J. Acoust. Soc. Amer.*, vol. 115, no. 2, pp. 739–756, 2004.
- [21] L. Savioja, J. Huopaniemi, and T. Lokki, “Creating interactive virtual acoustic environments,” *J. Audio Eng. Soc.*, vol. 47, no. 9, pp. 675–705, 1999.
- [22] N. Tsingos, T. Funkhouser, A. Ngan, and I. Carlbom, “Modeling acoustics in virtual environments using the uniform theory of diffraction,” in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 2001, pp. 545–552.
- [23] L. Cremer and H. A. Muller, *Principles and Applications of Room Acoustics*. London, U.K.: Applied Science, 1978, vol. 1.
- [24] D. R. Begault, *3D Sound for Virtual Reality and Multimedia*. Boston, MA: Academic, 1994.
- [25] W. Ahnert and R. Feistel, “EARS auralization software,” *J. Audio Eng. Soc.*, vol. 41, no. 1, pp. 894–904, 1993.
- [26] J.-M. Jot, L. Cerveau, and O. Warusfel, “Analysis and synthesis of room reverberation based on a statistical time-frequency model,” in *Proc. 103rd Audio Eng. Soc. Conv.*, 1997, preprint 4629.
- [27] J. F. Abbott, “The Interaction of Sound and Shock Waves With Flexible Porous Materials,” Ph.D. dissertation, Dept. Physics, Mass. Inst. Technol., Cambridge, MA, 1991.
- [28] *American National Standard Method for the Calculation of the Absorption of Sound by the Atmosphere*, ANSI SI.26-1978, American Institute of Physics (for Acoustical Society of America), 1978, New York.
- [29] W. G. Gardner, “The Virtual Acoustic Room,” M.S. thesis, Dept. Comput. Sci. Eng., Mass. Inst. Technol., Cambridge, MA, 1992.
- [30] N. Tsingos, I. Carlbom, G. Elko, R. Kubli, and T. Funkhouser, “Validating acoustical simulations in the Bells Labs box,” *IEEE Comput. Graph. Applicat.*, vol. 22, no. 4, pp. 28–37, 2002.
- [31] E.-E. Jan, P. Svaizer, and J. L. Flanagan, “Matched-filter processing of microphone array for spatial volume selectivity,” in *Proc. IEEE ISCAS*, Seattle, WA, 1995, pp. 1460–1463.
- [32] R. J. Renomeron, D. V. Rabinkin, J. C. French, and J. L. Flanagan, “Small-scale matched filter array processing for spatially selective sound capture,” (abstract), *J. Acoust. Soc. Amer.*, vol. 105, no. 5, p. 3208, (Proc. 143th JASA meeting, Dec. 1997).
- [33] D. V. Rabinkin, R. J. Renomeron, J. L. Flanagan, and D. F. Macomber, “Optimal truncation time for matched filter array processing,” in *Proc. IEEE ICASSP*, Seattle, WA, 1998, vol. 6, pp. 3629–3632.
- [34] P. M. Morse and K. U. Ingard, *Theoretical Acoustics*. Princeton, NJ: Princeton Univ. Press, 1968.
- [35] V. Rokhlin, “Rapid solution of integral equations of classical potential theory,” *J. Comput. Phys.*, vol. 60, no. 2, pp. 187–207, 1985.
- [36] T. Sakuma and Y. Yasuda, “Fast multipole boundary element method for large-scale steady-state sound field analysis. Part I: Setup and validation,” *Acustica/Acta Acustica*, vol. 88, no. 4, pp. 513–525, 2002.

- [37] Y. Yasuda and T. Sakuma, "Fast multipole boundary element method for large-scale steady-state sound field analysis. Part II: Examination of numerical items," *Acustica/Acta Acustica*, vol. 89, no. 1, pp. 28–38, 2003.
- [38] N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*. Amsterdam, The Netherlands: Elsevier Science, 2004.
- [39] M. Kleiner, B.-I. Dalenback, and P. Svensson, "Auralization—An overview," *J. Audio Eng. Soc.*, vol. 41, no. 11, pp. 861–875, 1993.
- [40] M. Vorlander, "International round robin on room acoustical computer simulations," in *Proc. 15th Int. Congr. Acoustics*, Trondheim, Norway, June 1995, pp. 689–692.
- [41] I. Bork, "A comparison of room simulation software—The 2nd round robin on room acoustical computer simulation," *Acustica/Acta Acustica*, vol. 86, no. 6, pp. 943–956, 2000.
- [42] A. Krokstad, S. Strom, and S. Sorsdal, "Calculating the acoustical room response by the use of a ray tracing technique," *J. Sound Vibr.*, vol. 8, no. 1, pp. 118–125, 1968.
- [43] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. Washington, D.C.: Nat. Bur. Stand., 1964.
- [44] R. Coifman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propagat. Mag.*, vol. 35, no. 3, pp. 7–12, Jun. 1993.
- [45] D. van Maercke and J. Martin, "The prediction of echograms and impulse responses within the Epidaure software," *Appl. Acoust.*, vol. 38, no. 2–4, pp. 93–114, 1993.
- [46] S. Zhang and J. M. Jin, *Computation of Special Functions*. New York: Wiley, 1996.



Ramani Duraiswami (M'99) received the Ph.D. degree from The Johns Hopkins University, Baltimore, MD, in 1991 and the B.Tech. degree from the Indian Institute of Technology, Bombay, in 1985.

He is a faculty member in the Department of Computer Science and the Institute for Advanced Computer Studies, University of Maryland, College Park, where he directs research at the Perceptual Interfaces and Reality Laboratory. His current research interests can broadly be categorized as audio for virtual reality and human computer interaction, scientific computing

(with a recent intense focus on the fast multipole method), and machine learning. He is an Associate Editor of the *ACM Transactions on Applied Perception*. More information on his research and publications can be obtained from <http://www.umiacs.umd.edu/~ramani>.

Dr. Duraiswami is a Member of the Audio and Electroacoustics Technical Committee of the IEEE Signal Processing Society.



Dmitry N. Zotkin was born in Moscow, Russia, in 1973. He received the combined B.S./M.S. degree in applied mathematics and physics from the Moscow Institute of Physics and Technology, Moscow, Russia, in 1996, and the M.S. and Ph.D. degrees in computer science from University of Maryland, College Park, in 1999 and 2002, respectively.

He is currently an Assistant Research Scientist at the Perceptual Interfaces and Reality Laboratory, Institute for Advanced Computer Studies (UMIACS), University of Maryland. His current research interests are in multichannel and microphone array signal processing for sound source localization and tracking in reverberant environments. He is also working in the broad areas of spatial audio and neuromorphic sound processing, including virtual auditory scene synthesis and customization, perceptual processing interfaces, biomimetic signal processing and acoustic pattern recognition, and associated problems.



Nail A. Gumerov received the M.S. degree in 1981 and the Ph.D. degree in 1987 from the Department of Mechanics and Mathematics, Lomonosov's Moscow State University. In 1992 he defended his Sc.D. dissertation (the highest scientific degree in the USSR and Russia) in mechanics of liquids, gas, and plasma.

From 1981 to 1993, he worked in several institutes of Russian Academy of Sciences in the field of Fluid Mechanics, Thermophysics, and Mechanics of Multiphase Systems and held positions from Research Fellow to the Principal Research Scientist. He also

worked as a Full Professor in 1996 in Bashkir State University, Russia. Since 1993, he has lived and worked in the U.S. In 2000, he joined the Institute for Advanced Computer Studies (UMIACS), University of Maryland at College Park, where he currently holds the position of Associate Research Scientist. His current research interests include scientific computing, efficient algorithms, mathematical modeling, acoustics and electromagnetics, fluid mechanics, and applied mathematics. In 2004 he wrote and published a book on applications of fast multipole methods to solution of the Helmholtz equation in three dimensions (in collaboration with Dr. R. Duraiswami).