Visibility-Based Beam Tracing for Soundfield Rendering

D. Marković, A. Canclini, F. Antonacci, A. Sarti, S. Tubaro

Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci, 32, 20133 Milano, Italy
dmarkovic/canclini/antonacc/sarti/tubaro@elet.polimi.it

***Abstract*—In this paper we present a visibility-based beam tracing solution for the simulation of the acoustics of environment that makes use of a projective geometry representation. More specifically, projective geometry turns out to be useful for the pre-computation of the visibility among all the reflectors in the environment. The simulation engine has a straightforward application in the rendering of the acoustics of virtual environments using loudspeaker arrays. More specifically, the acoustic wavefield is conceived as a superposition of acoustic beams, whose parameters (i.e. origin, orientation and aperture) are computed using the fast beam tracing methodology presented here. This information is processed by the rendering engine to compute spatial filters to be applied to the loudspeakers within the array. Simulative results show that an accurate simulation of the acoustic wavefield can be obtained using this approach.**

I. Introduction

In the past few years the spatial audio rendering has captured the attention of an ever-growing number of researchers and companies in the audio realm. New technological solutions have begun to appear in the literature and numerous commercial products are beginning to employ space-time processing for a wide range of applications. There is, in particular, a growing interest in the spatial rendering of acoustic sources in virtual environments using loudspeaker arrays, especially for applications of headset-free immersive telepresence and entertainment. In the first case, for example, the interest is in enabling virtual meetings where users would be able to exploit spatial rendering to carry out multiple cross-conversations at once and make the meeting more natural in its conduction. All such applications, however, rely on the synthesis or the reconstruction of the whole acoustic wavefield, which generally requires intensive computation. An efficient rendering solution is therefore needed, which exhibits sufficient flexibility to adapt to realistic conditions such as moving sources, arbitrarily shaped environments, etc.

The spatial impression of an individual source can be reproduced by adopting any of the rendering techniques that are available in the literature. One approach is offered by WaveField Synthesis [1], which exploits arrays of speakers; but we can also apply Ambisonics [2] [3] to a set of distributed speakers. An alternative array-based solution was recently proposed in [4], which enables the reconstruction of an arbitrary beampattern in a specific region in space. If we are interested in rendering not just a single source but also the impression of the whole surrounding environment, we could argue that we can approximate the whole wavefield as the superposition of the fields produced by the real source as well as by the corresponding set of image (mirrored by the reflectors) sources. This, however, turns out to be not enough, as the interactivity of the rendering experience heavily relies on the possibility to spatially "explore" the acoustic scene by moving in it. In doing so, we experience occlusions of virtual sources as a natural way to understand and interact with the environment. An extreme example of this is the ability to navigate in complex environments on the part of vision-impaired people.

One way to render a source in a complex environment could be to render all (real and virtual) sources, each with a beampattern that incorporates its visibility from the rendering region. This means that the natural beampattern of the source must be angularly windowed by the visibility conditions. In order to implement all that, we need a source renderer that is able to reproduce a source along with its beampattern [1], [2], [3], [4]. This renderer, however, requires a powerful modeling engine that is able to compute the location of sources and the shape of the visibility beams. This engine needs to be designed in such a way to determine the location and the visibility of sources in complex environments, to rapidly update such information as the source moves, and to keep track of the impact of wall reflection onto the acoustic contribution of each one of the sources. All this needs to be implemented without giving up a certain flexibility (accommodating moving sources) and efficiency (real-time operation). These requirements are typical of interactive multimedia applications but tend to severely limit the range of possible solutions that can be adopted for this goal.

Keeping track of visibility for a whole rendering region along with the location of the virtual sources is a feature that is not usually offered by geometric modeling methods. Such methods, in fact, tend to perform visibility checks only between the virtual source and a specific receiver location. Among the numerous path tracing solutions, however, there

MMSP'10, October 4-6, 2010, Saint-Malo, France.
978-1-4244-8112-5/10/$26.00

is one method that geometrically computes the visibility of sources from an entire region, which is beam tracing [5], [6], [7]. This method was conceived for tracing acoustic paths while overcoming the typical problems of spatial aliasing that traditional path tracing algorithms exhibit, as it works with compact bundles of rays (beams) originating from the same (image) source. By tracking such beams as they split and branch in their propagation and interaction with the environment, a data structure called beam tree is generated and looked up for an immediate determination of the acoustic paths. As beams are "cones" of visibility, the beam tree can be seen as a compact description of the visibility of a source from a whole spatial region.

A further generalization of this approach is found in [8], where the construction of the beam tree is derived from visibility information among all the reflectors in the environment. This information depends on the environment only, and can therefore be computed in advance. Once the source location is specified, the beam tree can be readily constructed from the visibility data structure through an iterative lookup process.

In this paper we start from the modeling engine proposed in [8] and we reformulate its development in order to make it suitable for rendering purposes. In particular, a new parametrization based on homogeneous coordinates is adopted in order to simplify the management of visibility information and to make it reflector-independent. This new formulation has a twofold advantage: the visibility data structure becomes more compact; and reflections become linear projective transformations.

This modeling engine is here paired with a multiple source renderer that we first presented in [9], based on an earlier work in [4], based on a speaker array. This 2D renderer represents a good match with our modeling engine, thanks to its parametric compatibility and effectiveness for rendering purposes. Simulation results of wavefield reconstruction based on this structure are provided.

The rest of the paper is structured as follows: Section II describes the proposed beam tracing algorithm. Section III illustrates the approach used for soundfield rendering with an array of loudspeakers. Section IV provides experimental results for wavefield simulation.

II. Modeling the soundfield of the environment

In this Section we first introduce the ray parameter space. We do so using the oriented projective geometry [10] that allows us to describe oriented lines and planes that otherwise could not be handled using classical projective geometry. Then we map a minimal set of elementary objects that are sufficient to characterize the propagation in the environment into the ray parameter space. In particular, the geometric primitives we are interested in are: acoustic rays; sources and receivers; reflectors; and beams. This information is sufficient to characterize the visibility information. With this information at hand, beam tracing becomes a lookup on the visibility data structure. The outcome of the beam tracing step is the beam tree. As discussed above, in the rendering context we are interested in the visibility of the virtual sources from a whole region. As a consequence beams are sufficient for our goal. Nonetheless, we briefly present also the computation of acoustic paths using the beam tree.

A. Transformations from geometric to ray space

1) The ray parameter space: The acoustic ray can be seen as an oriented line in the geometric space. A line in $\mathbb{R}^2$ is represented by the equation

$$l_1 x_1 + l_2 x_2 + l_3 = 0.$$

We parameterize a ray with the coordinates $[l_1, l_2, l_3]^T$ of the line on which the ray lies. We notice that the vectors $[l_1, l_2, l_3]^T$ and $k[l_1, l_2, l_3]^T, k \in \mathbb{R}, k \neq 0$ represent the same ray. As a consequence, this parametrization defines a class of equivalence, as it uses scalable - homogeneous - coordinates. However, rays have a travel direction. In order to distinguish rays lying on the same line but with opposite orientations, we limit the range of the scalar k to the positive or negative interval, as common in the Oriented Projective Geometry (OPG) [10]

$$\mathbf{l}_1 = k[l_1, l_2, l_3]^T, k > 0$$
$$\mathbf{l}_2 = k[l_1, l_2, l_3]^T, k < 0\ .$$

A generic point in the (l_1, l_2, l_3) space corresponds to a ray in the geometric space and thus this parametrization is here referred as the ray space. The equivalence class inherent in the

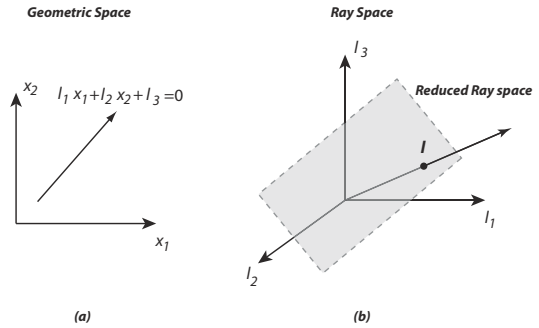


Figure 1. A ray in geometric (a) and ray spaces (b).

ray space implies that the ray space is a projective space $\mathbb{P}^2$, which is essentially $\mathbb{R}^3$ without $(0, 0, 0)$. For clarity of visualization, rather than visualizing the whole three dimensional ray space, we depict the primitives in a reduced 2D ray space, obtained by intersecting the ray space with a prescribed plane, as shown in Fig. 1 (b). We notice, however, that in the reduced ray space we cannot distinguish rays with the same direction but with opposite orientations.

2) Acoustic source and receiver: Sources and receivers can be seen as points in the geometric space. We represent a point $P = (x_1, x_2)$ in the ray space with the set of all rays that pass through it. The homogeneous coordinates of the point P are $\mathbf{x}_P = k[x_1, x_2, 1]^T, \quad k > 0$. The set of rays passing through P is

$$\overline{P} = \{(l_1, l_2, l_3) \in \mathbb{R}^3 | l_1 x_1 + l_2 x_2 + l_3 = 0\} = \{\mathbf{l} \in \mathbb{P}^2 | \mathbf{x}_P^T \mathbf{l} = 0\}. \quad (1)$$

Note that the equation in (1) defines in the ray space a plane passing through the origin, as $\mathbf{x}_P$ is known. An example of the representation of a point in the geometric and reduced ray spaces is in Fig. 3(b).

3) Reflector: In the geometric domain the reflector R is a line segment and it is completely defined by the two endpoints $A = ({x_1}^A, {x_2}^A)$ and $B = ({x_1}^B, {x_2}^B)$. As for points, we represent the reflector in the ray space as the set of all rays that pass through it, i.e. through all the intermediate points. In the ray space this corresponds to the set of all planes representing the infinite intermediate points between A and B.

$$\overline{R} = \overline{A} \cup ... \cup \overline{P_i} \cup ... \cup \overline{B}.$$

In accordance with the image source principle, when we evaluate the visibility of the environment from a mirrored source, we do not consider the reflectors in the half-space where the mirrored source lies, as illustrated in Figure 2. This motivates us in defining two reflectors, one for each face of the line segment. Traditional projective geometry does not account for oriented reflectors, while Oriented Projective Geometry provides the tools required for representing them. With reference to Fig.3(c), the two rays $\mathbf{l}'$ and $\mathbf{l}''$, which fall

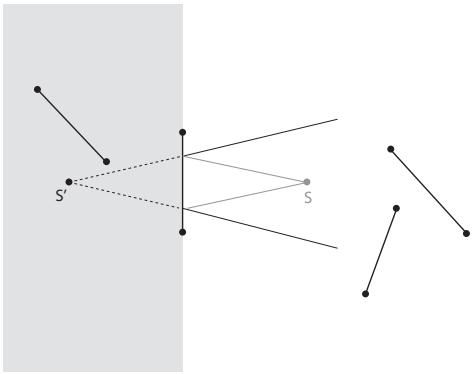


Figure 2. The source S is mirrored over the reflector in order to obtain the source S'. According to the image source principle, the obstacles in the half-space in which S' lies (depicted with the shaded wiring) are not considered in the visibility evaluation from S'.

onto the opposite faces of the line segment R in the travel direction have the endpoints A and B on opposite sides: A is on the left for $\mathbf{l}'$ and on the right for $\mathbf{l}''$ and therefore $\mathbf{x}_A^T\mathbf{l}' > 0$ and $\mathbf{x}_A^T\mathbf{l}'' < 0$. We exploit this inequalities when we represent the two oriented reflectors $\overline{R}_1$ and $\overline{R}_2$ corresponding to the non-oriented reflector $\overline{R}$ in the ray space:

$$\overline{R}_1 = \{\mathbf{l} \in \mathbb{P}^2 | \mathbf{x}_A^T\mathbf{l} > 0\} \cap \{\mathbf{l} \in \mathbb{P}^2 | \mathbf{x}_B^T\mathbf{l} < 0\} = \overline{A}_+ \cap \overline{B}_-,$$

$$\overline{R}_2 = \{\mathbf{l} \in \mathbb{P}^2 | \mathbf{x}_A^T\mathbf{l} < 0\} \cap \{\mathbf{l} \in \mathbb{P}^2 | \mathbf{x}_B^T\mathbf{l} > 0\} = \overline{A}_- \cap \overline{B}_+ = -\overline{R_1}..$$

The non oriented reflector can be expressed in a closed form as the union of two oriented reflectors that compose it

$$\overline{R} = \overline{R}_1 \cup \overline{R}_2 = \{\overline{A}_+ \cap \overline{B}_-\} \cup \{\overline{A}_- \cap \overline{B}_+\}.$$

4) Visibility region: All the rays originated from an oriented reflector R_i form the region of visibility from that reflector

$$\mathcal{R}(R_i) = -\overline{R}_i.$$

By intersecting this region with the rays that fall onto another oriented reflector R_j we obtain the visibility region of R_j from R_i:

$$\mathcal{V}(R_i, R_j) = \mathcal{R}(R_i) \cap \overline{R}_j = (\overline{A}_{+/-} \cap \overline{B}_{-/+}) \cap (\overline{C}_{+/-} \cap \overline{D}_{-/+}).$$

With reference to Fig. 3(d), in the reduced ray space the visibility region $\mathcal{V}(R_i, R_j)$ is given by the intersection of four half spaces that form a pyramid with the apex at the origin of the ray space.

5) Visibility diagram: If the environment is composed of more than two reflectors, mutual occlusions arise. This corresponds to an overlapping of visibility regions in the ray space. Sorting out which reflector occludes which in the geometric space means determining which visibility region overrides which in their overlap as in Fig. 3(e). We do this casting a test ray in the overlap region and finding which reflector this ray falls first, as illustrated in Fig.3(e). The resulting collection of visibility regions constitutes the visibility diagram of the reflector R_i

$$\mathcal{D}(R_i) = \{\mathcal{V}^*(R_i, R_j) \neq \emptyset\},$$

where R_j are reflectors visible from R_i and $*$ indicates that visibility regions have been overridden according to the front-to-back order.

The collection of visibility diagrams of all the reflectors determines the *global visibility*. Notice that in order to build visibility diagrams we do not need the knowledge of source and receiver positions and thus the evaluation of the mutual visibility between reflectors can be done in an off-line mode.

In [8] the ray parameter space was defined with a transformation of the geometric space that was reflector-dependent. As a consequence, the ray parameter space depends on the geometric transformation. Conversely, the parametrization adopted here is reflector-independent and the visibility diagram of different reflectors are composed by the same visibility regions overridden with different front-to-back order. The new parametrization improves the efficiency of the representation since it does not require the above geometric transformation.

6) Beam: The beam $\mathbf{b}$ is completely specified by an origin (virtual source) and by the (connected) illuminated region of the reflector it falls onto. In the ray space this corresponds to intersecting the representations of the virtual source and of the illuminated portion of the reflector, as shown in Fig. 3(f):

$$\mathbf{b} = \overline{S} \cap \overline{R_i}|_{\mathbf{l}'}^{\mathbf{l}''}.$$

B. Beam tracing

With reference to Fig.4, we show how we use visibility diagrams to iteratively trace beams. Let us consider the reflection of a beam $\mathbf{b}_i$ onto the reflector AB. We first compute

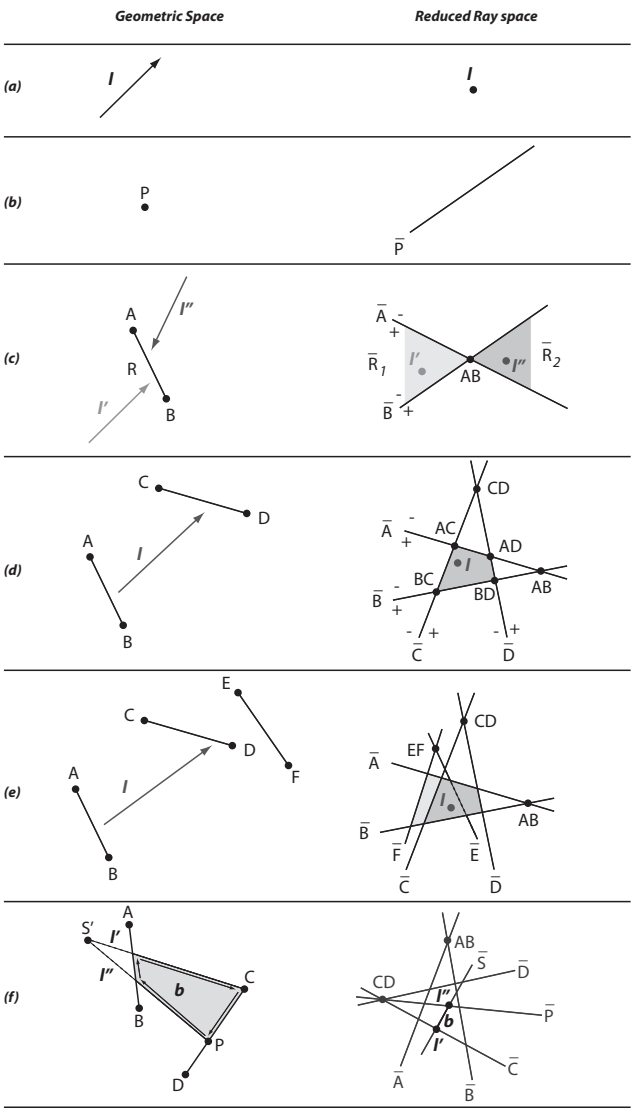


Figure 3. Transformations from geometric to ray space: (a) ray; (b) point; (c) reflector; (d) visibility region; (e) visibility diagram; (f) beam.

the reflected bundle of rays $\mathbf{b}'_i$ finding the virtual source S' (determined by mirroring the source S with respect to segment AB) and rays $\mathbf{l}'$ and $\mathbf{l}''$ that limit the reflected bundle of rays $\mathbf{b}'_i$. The splitting process is accomplished in the ray space by intersecting the reflector's visibility diagram with the ray space representation of $\mathbf{b}'_i$ (i.e. the portion of the plane $\overline{S'}$ limited by $\mathbf{l}'$ and $\mathbf{l}''$) as shown in Fig. 4(b). The ray space representation of $\mathbf{b}'_i$ is made of a number of segments, each lying in a different visibility region. These segments represent the sub-beams originated from the splitting of $\mathbf{b}'_i$. The corresponding beams in the geometric space are traced in Fig. 4(c). Some of them proceed to infinity, others are blocked by reflectors and therefore they originate new beams. The recursive procedure stops when the beam tree reaches a preassigned order of reflection or when the beams die out (i.e., when they are attenuated below a preassigned threshold of magnitude). The beams are organized in the beam tree

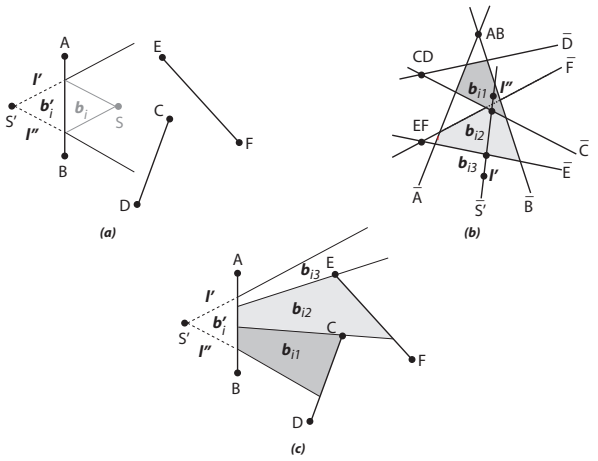


Figure 4. (a) Reflection of the beam in the geometric space; (b) Beam subdivision in the ray space; (c) New beams in the geometric space.

data structure that contains the branching relationship between acoustic beams and represent efficiently the visibility from the geometric space of the source position. This representation turns out to be useful for the rendering stage.

C. Path tracing

Once the receiver location is specified, a simple iterative procedure looks up the beam tree to find the paths from source to receiver. We use the beam parametrization shown in Fig. 3(f), where the beam is parameterized with the vectors corresponding to its bounding lines and they are oriented in order to guarantee that a point inside the beam is always on the right of those vectors. In order to test if a point is inside the beam, therefore, we only need to verify that the receiver is on the right side of all vectors that parameterize the beam.

III. Soundfield rendering

The beam tree contains all the information that we need to structure a soundfield as a superposition of individual beams. We now consider the problem of rendering the acoustics of a virtual environment. Individual beams can be reconstructed using an array of loudspeakers whose spatial filters are designed by minimizing the difference between the desired and the emitted soundfields as described in [4]. Using the beam superposition principle the rendering of the overall soundfield is achieved by simply adding together the spatial filters for the individual beams. In the next paragraphs we summarize this approach.

A. Rendering an acoustic beam

We use an array of M loudspeakers in positions $\mathbf{p}_m$, $m = 1, ..., M$, to simulate the arbitrary beampattern of a virtual source $\mathbf{s}$. We define N test points $\mathbf{a}_n$, $\mathbf{n} = 1, ..., N$, in a listening area of arbitrary shape (Fig. 5).

We impose that the wavefield on the test points best approximates the wavefield produced by the virtual source

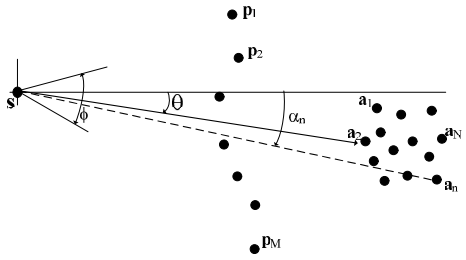


Figure 5. The virtual source is located in $\mathbf{s}$ and emits a beam towards the direction θ with angular aperture ϕ; the points $\mathbf{p}_1$,...,$\mathbf{p}_M$ denote the loudspeaker array; $\mathbf{a}_1$,...,$\mathbf{a}_N$ are the test points.

with the specific beampattern obtaining the following matrix formulation:

$$\mathbf{G}\mathbf{h} = \mathbf{r}_d,$$

where $\mathbf{r}_d$ is the desired response, $\mathbf{G}$ is the $N \times M$ propagation matrix from each loudspeaker to each test point and $\mathbf{h}$ is the spatial filter of the loudspeaker array (i.e. the vector of complex coefficients applied to the loudspeakers).

Usually $N \gg M$, i.e. the system is overdetermined, and it admits no exact solution. Using the inverse problems theory, an estimation $\hat{\mathbf{h}}$ of $\mathbf{h}$ can be calculated using the pseudo-inverse of the matrix $\mathbf{G}$, $\mathbf{G}^+$. The loudspeakers weight vector is approximated by:

$$\hat{\mathbf{h}} = \mathbf{G}^+\mathbf{r}_d = (\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H\mathbf{r}_d,$$

which represents the best solution to the problem in the least square sense.

The matrix $\mathbf{G}^H\mathbf{G}$ is positive definite and, therefore invertible, but it may be ill-conditioned. In order to avoid instability problems a reconditioning of $\mathbf{G}^H\mathbf{G}$ is needed. We do so through Singular Value Decomposition ($\mathbf{G}^H\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$) retaining the columns that guarantee an acceptable value of the conditioning number $\sigma_{min}/\sigma_{max}$, where σ_i is a singular value.

B. Extension to multiple image sources and wideband signals

The extension for the rendering of the overall soundfield defined as a superposition of beams is done by summing up the loudspeaker weights for all the individual beams found in the beam tracing procedure

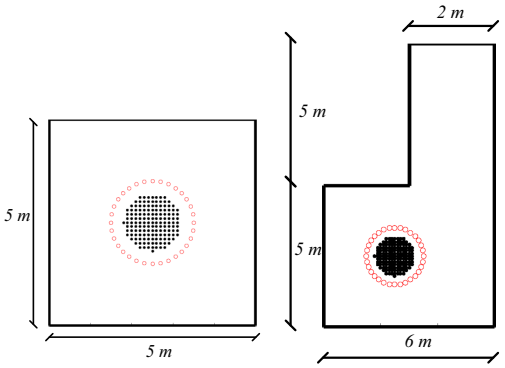
$$\hat{\mathbf{h}}_{TOT} = \sum_{z=1}^{Z} \hat{\mathbf{h}}_z = \sum_{z=1}^{Z} \mathbf{G}_z{}^+\mathbf{r}_{d_z}.$$

The extension to wideband signals is done by estimating a filter for each loudspeaker through a frequency sampling approach. We compute the spatial filters $F_m(f_l), m = 1, \ldots, M$ for the frequencies $f_1, \ldots, f_L$, where L is the number of controlled frequencies. We proceed, then, to an interpolation stage that computes the spatial filters $F_m(f_k), m = 1, \ldots, M$ from $F_m(f_l)$. The frequencies f_k are disposed on a regularly spaced frequency grid of K elements. In particular, the interpolation processing is parabolic for the amplitude and cubic for the phase. As a result, we obtain a set of filters that are fed with the signal to be rendered. In order to preserve the spatial Nyquist criterion the maximum operating frequency is limited to $f_{max} < \frac{c}{2d}$, where d is the distance between loudspeakers.

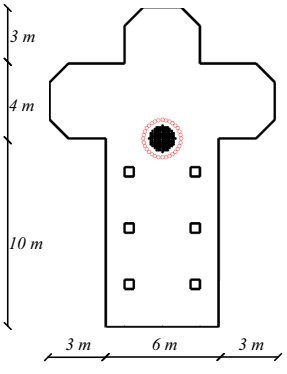
IV. Experimental results

In this Section we show some simulations in order to illustrate the accuracy of the rendering. The experimental set-up consists of a circular array with radius $R = 1m$ composed by $M = 32$ equally spaced loudspeakers and enclosing a listening area made by $N = 149$ points uniformly distributed in a circular region with radius $r = 0.7m$ concentric with the loudspeakers array. The reflection coefficient of the walls in the simulations is 0.7. We stopped the beam tracing at the fourth order of reflection. The source is always located in the center of the controlled area. If we superimpose the direct wavefront with the reverberated soundfield, a masking effect appears, which attenuates the difference between the desired and simulated soundfields. Thus, we show in the simulations only the soundfield created by the reflected acoustic beams.

We simulate the acoustics of three different virtual environments, shown in Figures 6(a)-6(c). In the environment in (a), all walls are mutually visible. The environment in (b) exhibits some mutual occlusions among walls. Finally, the environment in (c) is a small church with a number of possible occlusions. The metric we use to evaluate the accuracy of the rendered



(a) Test Environment 1 (b) Test Environment 2



(c) Test Environment 3

Figure 6. Test environments. The red circles represent the loudspeakers. The area enclosed by the black points represent the area in which the rendering is performed.

soundfield is the *Root Mean Square Error (RMSE)* between the predicted and simulated soundfields. For narrowband sources the RMSE is defined as

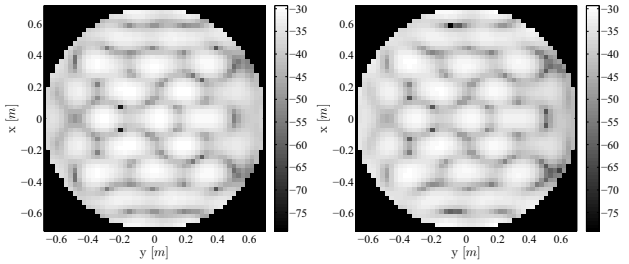
$$E_{RMSE} = \sqrt{\frac{\sum_{i=1}^{Q}[S(\mathbf{q}_i) - \hat{S}(\mathbf{q}_i)]^2}{Q}}\,, \qquad (2)$$

where Q is the number of points that compose the soundfield image, S is the desired soundfield and $\hat{S}$ is the simulated soundfield. For wideband sources the RMSE is defined as

$$E_{RMSE} = \sqrt{\frac{\sum_{k=1}^{K}\sum_{i=1}^{Q}[S(\mathbf{q}_i, f_k) - \hat{S}(\mathbf{q}_i, f_k)]^2}{QK}}\,, \qquad (3)$$

where k is the frequency index of the FFT. For wideband sources the signal is a speech filtered in the band $[100Hz, 2kHz]$.

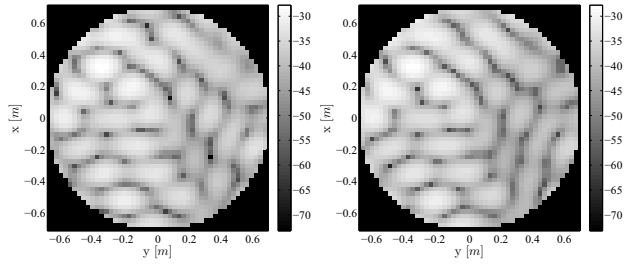
Figures 7(a) and 7(b) show the desired and simulated soundfields, respectively, when the source is narrowband at 1kHz. The resulting RMSE is 2.57dB. When we perform the same experiment for wideband sources the RMSE is 3.31dB.



(a) Desired soundfield (b) Simulated soundfield

Figure 7. Desired and rendered soundfields for the environment in Fig.6(a). The source is narrowband at 1kHz. The resulting RMSE is 2.57dB.

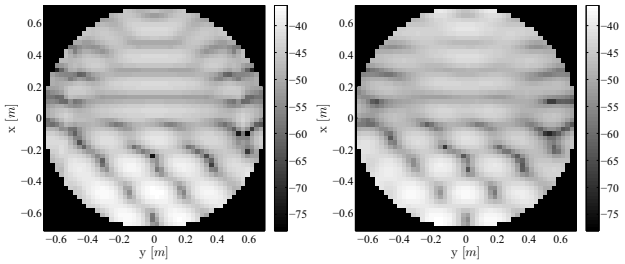
Figures 8(a) and 8(b) show the desired and rendered soundfields for the second environment. In this case the RMSE is 2.81dB, while for wideband sources is 3.96dB.



(a) Desired soundfield (b) Simulated soundfield

Figure 8. Desired and rendered soundfields for the environment in Fig.6(b). The source is narrowband at 1kHz. The resulting RMSE is 2.81 dB

Finally, Figures 9(a) and 9(b) show the predicted and simulated soundfields for the third environment. Even if the environment is more complex that the previous ones, we notice a good match between the soundfields. If we repeat the same experiment with wideband sources the resulting RMSE is 6dB. As a general consideration, we can observe that in all the three cases the soundfield has been correctly reproduced, except for some restricted spots, where the predicted soundfield exhibits some subtleties that the beam rendering methodology is not able to reproduce.



(a) Desired soundfield (b) Simulated soundfield

Figure 9. Desired and rendered soundfields for the environment in Fig.6(c). The source is narrowband at 1kHz. The resulting RMSE is 2.31 dB

V. Conclusions

In this paper we showed the application of a fast beam tracing algorithm to render the acoustics of virtual environments. The beam tracing algorithm presented here is a generalization of the algorithm in [8], which turns out to be useful for the beam rendering engine. Results of wavefield simulations for three different virtual environments show a good match between predicted and simulated soundfields. [1]

References

[1] A. Berkhout, D. D. Vries, and P. Vogel, "Acoustic control by wave field synthesis," *J.Acoust.Soc.Am.*, vol. 93, pp. 2764–2778, 1993.

[2] M. A. Gerzon, "Practical periphony: The reproduction of full-sphere sound," in *Preprint 65th Conv. Aud. Eng. Soc.*, Feb. 1980.

[3] D. Menzies and M. Al-Akaidi, "Nearfield binaural synthesis and ambisonics," *The Journal of the Acoustical Society of America*, vol. 121, no. 3, pp. 1559–1563, 2007.

[4] A. Canclini, A. Galbiati, A. Calatroni, F. Antonacci, A. Sarti, and S. Tubaro, "Rendering of an acoustic beam through an array of loudspeakers," in *Proceedings of the 12th International Conference on Digital Audio Effects*, DAFx-09, Como, Italy, September 2009.

[5] J. P. Walsh and N. Dadoun, "What are we waiting for? the development of godot," in *Proceedings of 103rd Meeting of the Acoustical Society of America*, vol. 71, no. S1, New York, April 1982, p. S5.

[6] T. Funkhouser, I. Carlbom, G. Elko, G. Pingali, M. Sondhi, and J. West, "Beam tracing approach to acoustic modeling for interactive virtual environments," in *Proceedings of the ACM Conference on Computer Graphics*, SIGGRAPH 98, Orlando, Florida, July 1998, pp. 21–32.

[7] T. Funkhouser, P. Min, and I. Carlbom, "Real-time acoustic modeling for distributed virtual environments," in *Proceedings of the ACM Conference on Computer Graphics*, SIGGRAPH 99, Los Angeles, California, 1999, pp. 365–374.

[8] F. Antonacci, M. Foco, A. Sarti, and S. Tubaro, "Fast tracing of acoustic beams and paths through visibility lookup," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 4, pp. 812–824, May 2008.

[9] F. Antonacci, A. Calatroni, A. Canclini, A. Galbiati, A. Sarti, and S. Tubaro, "Soundfield rendering with loudspeaker arrays through multiple beam shaping," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, WASPAA'09, New Paltz, New York, October 2009, pp. 313–316.

[10] J. Stolfi, *Oriented Projective Geometry: A Framework for Geometric Computations*. Academic Press, 1991.

[1]The work is realized under the funding of the SCENIC project. The SCENIC project acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 226007