計算数理工学論文集 Vol. 21 (2021 年 12 月)，論文 No. 19-211218 JASCOME

遠方近似を用いた3次元時間域境界要素法による仮想空間における移動点のリアルタイム可聴化

A Real-Time Auralization in a Virtual Space Using Time Domain Boundary Integral Equation Method with the Far Field Approximation

吉川　仁 [1)]，西上　駿祐 [2)]

Hitoshi YOSHIKAWA and Shunsuke NISHIGAMI

1) 京都大学大学院情報学研究科　(〒 606-8501　京都市左京区吉田本町,　E-mail: yskhit@i.kyoto-u.ac.jp)
2) 株式会社ギフティ　(〒 141-0022　東京都品川区東五反田 2-10-2,　E-mail:)

A numerical computation of a sound that a character moving in the 3D virtual space hears is considered. We consider wave acoustics for the accurate computation of the sound field having the several complicated scatterers. We solve the scattered wave field using time domain BIEM and store the computed boundary values in advance. We compute the sound of the moving point with the integral representation formula and the far field approximation in real time. The real-time auralization is realized with the game engine, Unity3D.

Key Words: game sound, real-time auralization, BEM, BIEM, Unity

1. はじめに

近年、バーチャルリアリティ技術の発展によりプレイヤー自身がバーチャル空間内を動き回るゲームや、プレイヤーが操作するキャラクターが 3 次元のオープンワールドを自由に移動できるゲームが数多く開発されている。ファーストパーソン・シューティングゲー (FPS) やサードパーソン・シューティングゲーム (TPS) などの対戦シューティングゲームにおいては、敵の発砲音を聞くことで敵の位置情報を得るテクニックなども存在する [(1)]。これらのゲームでは、バーチャル空間内を移動するキャラクターやプレイヤー自身に聞こえる音をより正確に再現することが求められている [(2)]。

樫山ら [(3, 4)] は幾何音響理論に基づく解析を行い、音響のパワーレベルを算出し、バーチャル空間でのリアルタイム可聴化を実現している。しかしながら、この手法では音の特性である波動性が失われてしまうことや、複雑な散乱体形状に対応しづらいといった問題点があり、正確な音場の再現には限界があった。Raghuvanshi ら [(5)] も音波の伝播経路を直達音と反射音に分けてリアルタイム可聴化を実現しているが、こちらも音の波動性を考慮しておらず、正確な音場を再現しているとは言い難い。このように、現在行われているバーチャル空間での音場の数値解析は、幾何音響シミュレーションを用いた簡易的な数値手法であり、波動性を考慮した正確な音場の再現が可能な数値解析が必要とされている [(6)]

そこで、本研究では、**Fig.1** に示すような 3 次元バーチャル空間内を移動するキャラクターに聞こえる音をリアルタイムに再現することを目的とする。なお、本論文における “リアルタイム ”の意味は、ゲーム内における実時間を意味する。つまり、キャタクターがバーチャル空間を移動するに合わせて、その位置での音を実時間で計算することを目的とする。

これまでに、時間域境界要素法 [(7)] を用いることで散乱体を 1 つ持つような単純な領域でのリアルタイム可聴化が実現されている。境界要素法には、領域の境界上での境界量を事前に計算しておけば、任意の内点での解は簡単なベクトル積演算を行うだけで求まるという特徴がある。この特徴を利用し、境界量と影響係数とのベクトル積演算のみをリアルタイムで実行することで計算時間を短縮し、散乱体を 1 つ持つような単純な領域でのリアルタイム可聴化を実現した [(8, 9)]。しかし、領域内に複数の散乱体がある場合には、ベクトルの次元数が大きくなりベクトル積演算でさえ計算時間が大きくなるため、依然リアルタイム可聴化には至っていない。そこで、本研究では、キャラクター位置から遠方にある散乱体からの影響に対しては近似を用いて計算することで計算時間を短縮し、複数の散乱体が存在する領域でのリアルタイム可聴化を試みる。

2021 年 11 月 4 日受付，2021 年 11 月 22 日受理

Fig. 1　バーチャル空間内を移動するキャラクター

2. 時間域境界積分方程式による音場解析

Fig.2 に示すような散乱体を持つある閉じた 3 次元領域の外部領域 D における、位置 $\boldsymbol{x}$、時刻 t での音圧 $u(\boldsymbol{x},t)$ について次の初期値境界値問題を考える。

$$\ddot{u}(\boldsymbol{x},t) - c^2 u_{,ii}(\boldsymbol{x},t) = 0, \quad \boldsymbol{x} \in D,\ t > 0 \tag{1}$$

$$u(\boldsymbol{x},0) = \dot{u}(\boldsymbol{x},0) = 0, \quad \boldsymbol{x} \in D \tag{2}$$

$$u(\boldsymbol{x},t) = \bar{u}(\boldsymbol{x},t), \quad \boldsymbol{x} \in S_D,\ t > 0 \tag{3}$$

$$\frac{\partial u}{\partial n}(\boldsymbol{x},t) = \bar{q}(\boldsymbol{x},t), \quad \boldsymbol{x} \in S_N,\ t > 0 \tag{4}$$

$$u(\boldsymbol{x},t) \to u^{\mathrm{in}}(\boldsymbol{x},t), \quad |\boldsymbol{x}| \to \infty \tag{5}$$

ここに、c は音速、$u^{\mathrm{in}}(\boldsymbol{x},t)$ は入射波、S は D の境界であり、境界 S の一部で Dirichlet 境界条件が課せられる境界を S_D、Neumann 境界条件が課せられる境界を S_N とする。なお、$S = S_D \cup S_N$ である。また、$\dot{(\ )}$ は時間に関する微分 $\dfrac{\partial}{\partial t}$ を、$(\ )_{,i}$ は空間に関する微分 $\dfrac{\partial}{\partial x_i}$ を表す。$\dfrac{\partial}{\partial n} = n_i(\boldsymbol{x})\dfrac{\partial}{\partial x_i}$ であり、$\boldsymbol{n}(\boldsymbol{x})$ は $\boldsymbol{x} \in S$ における領域 D より外に向いた単位法線ベクトルを表す。$\bar{u}(\boldsymbol{x},t)$, $\bar{q}(\boldsymbol{x},t)$ は既知関数である。本研究では、入射波 $u^{\mathrm{in}}(\boldsymbol{x},t)$ として、静止した 1 点の点音源 $\boldsymbol{x}^{\mathrm{o}}$ から発信される音 $u^{\mathrm{o}}(t)$ を考える。入射波 $u^{\mathrm{in}}(\boldsymbol{x},t)$ は、波速 c の球面波として次の様に与えられる。

$$u^{\mathrm{in}}(\boldsymbol{x},t) = \frac{u^{\mathrm{o}}(t - r_{\mathrm{o}}/c)}{4\pi r_{\mathrm{o}}}, \qquad r_{\mathrm{o}} = |\boldsymbol{x}^{\mathrm{o}} - \boldsymbol{x}| \tag{6}$$

2.1. 境界積分方程式法による境界量の計算

波動方程式の初期値境界値問題 (式 (1)〜(5)) より得られる境界積分方程式は次の通りである。

$$\begin{aligned}\frac{1}{2}u(\boldsymbol{x},t) =& u^{\mathrm{in}}(\boldsymbol{x},t)\\ &+ \int_S\int_0^t \Gamma(\boldsymbol{x}-\boldsymbol{y},t-s)\frac{\partial u}{\partial n}(\boldsymbol{y},s)dsdS\\ &- \int_S\int_0^t \frac{\partial \Gamma}{\partial n_y}(\boldsymbol{x}-\boldsymbol{y},t-s)u(\boldsymbol{y},s)dsdS,\\ &\qquad \boldsymbol{x} \in S,\ t > 0\end{aligned} \tag{7}$$

ここに、Γ は 3 次元波動方程式の基本解であり、

$$\Gamma(\boldsymbol{x},t) = \frac{\delta(t - |\boldsymbol{x}|/c)}{4\pi|\boldsymbol{x}|} \tag{8}$$

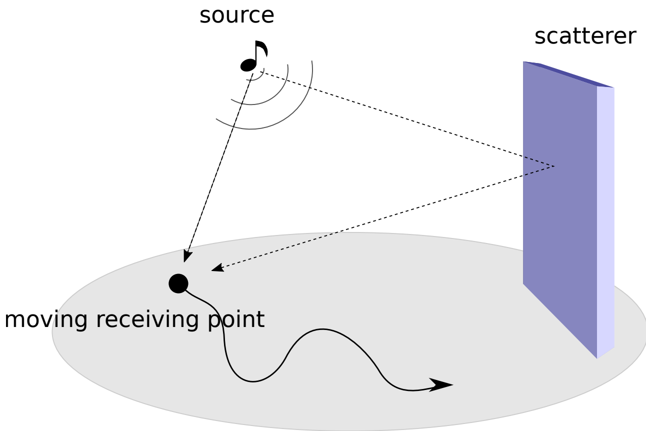

Fig. 2　点音源からの球面波を入射波とする散乱音場

である。ここに、$\delta(t)$ は Dirac のデルタ関数である。境界積分方程式 (7) を未知な境界値である $u(\boldsymbol{x},t)$, $\boldsymbol{x} \in S_N$ と $\dfrac{\partial u}{\partial n}(\boldsymbol{x},t)$, $\boldsymbol{x} \in S_D$ について解く。時間域の境界積分方程式法では、境界 S を複数の境界要素 S_j, $j = 1,\cdots,N$ に分割し、また時間刻み幅を Δt とし、境界積分方程式 (7) 右辺の積分を時間ステップ毎に数値的に計算し、未知境界量 $u(\boldsymbol{x}^j, n\Delta t), \boldsymbol{x}_j \in S_N$ と $\dfrac{\partial u}{\partial n}(\boldsymbol{x}^j, n\Delta t), \boldsymbol{x}_j \in S_D$, $n = 1,2,\cdots$ を求める。ここに、$\boldsymbol{x}^j$ は境界要素 S_j の代表点である。

2.2. 解の積分表現を用いた内点計算

境界積分方程式を解くことで、境界 S 上の $u(\boldsymbol{x},t)$, $\dfrac{\partial u}{\partial n}(\boldsymbol{x},t)$ が全て得られるため、領域 D 内の任意の点 $\boldsymbol{x}$ での解が、解の積分表現

$$\begin{aligned}u(\boldsymbol{x},t) =& u^{\mathrm{in}}(\boldsymbol{x},t)\\ &+ \int_S\int_0^t \Gamma(\boldsymbol{x}-\boldsymbol{y},t-s)\frac{\partial u}{\partial n}(\boldsymbol{y},s)dsdS\\ &- \int_S\int_0^t \frac{\partial \Gamma}{\partial n}(\boldsymbol{x}-\boldsymbol{y},t-s)u(\boldsymbol{y},s)dsdS,\\ &\qquad \boldsymbol{x} \in D,\ t > 0\end{aligned} \tag{9}$$

より求まる。この計算は内点計算とも呼ばれ、境界積分方程式法では、式 (9) 右辺の境界積分も数値的に計算する。

2.3. リアルタイム可聴化のための工夫

本研究において、式 (9) の内点 $\boldsymbol{x}$ はバーチャル空間内を動くキャラクターの位置を表す。キャラクターはバーチャル空間内を時間と共に連続的に動く。そのため、キャラクター位置と関係なく求められる境界量は式 (7) により事前に数値的に求めておき、内点計算のみをキャラクターの移動と共にリアルタイムで計算する。

さらに、例えば完全反射条件 ($\bar{q}(\boldsymbol{x},t) = 0$) のように境界条件が時間依存ではない場合について、ベクトル積演算量とメモリにストアしておく境界値の量を減らすために、次の手順で内点計算を行う [9]。以下では、式 (3),(4) において時間依存しない境界条件を考える。

$$\bar{u}(\boldsymbol{x},t) = \tilde{u}(\boldsymbol{x}) \tag{10}$$

$$\bar{q}(\boldsymbol{x},t) = \tilde{q}(\boldsymbol{x}) \tag{11}$$

1. まず、時間域の内挿関数 $M_T^m(t)$ を用いて点音源から発信される音 u^o を次の様に離散化する。

$$u^o(t) \simeq \sum_{m=1}^{N_T} u^o(m\Delta t)M_T^m(t) \tag{12}$$

ここに、発信音 $u^o(t)$ は $0 \leq t \leq N_T\Delta t$ の音データとする。

2. 次に、メモリにストアするデータ量を少なくするために、サポートが短い関数である M_T^1 を入射波とする次の初期値境界値問題の境界値を数値的に求めてメモリにストアする。

$$\ddot{u}^M(\boldsymbol{x},t) - c^2u^M_{,ii}(\boldsymbol{x},t) = 0, \quad \boldsymbol{x} \in D,\ t > 0 \tag{13}$$

$$u^M(\boldsymbol{x},0) = \dot{u}^M(\boldsymbol{x},0) = 0, \quad \boldsymbol{x} \in D \tag{14}$$

$$u^M(\boldsymbol{x},t) = \tilde{u}(\boldsymbol{x}), \quad \boldsymbol{x} \ \in \ S_D,\ t > 0 \tag{15}$$

$$\frac{\partial u^M}{\partial n}(\boldsymbol{x},t) = \tilde{q}(\boldsymbol{x}), \quad \boldsymbol{x} \ \in \ S,\ t > 0 \tag{16}$$

$$u^M(\boldsymbol{x},t) \to \frac{{M_T}^1(t - r_o/c)}{4\pi r_o}, \quad |\boldsymbol{x}| \to \infty \tag{17}$$

なお、Δt は一定であり、また考えている問題は線形の問題であるため、解の重ね合わせの原理より式 (9) 右辺の時間積分を離散化して求まる数値解 $u(\boldsymbol{x}, n\Delta t)$ は、式 (13)〜式 (17) に示した初期値境界値問題の解 $u^M(\boldsymbol{x}, n\Delta t)$ を用いて、次の畳み込みの形で計算できる。

$$u(\boldsymbol{x}, n\Delta t) = \sum_{m=1}^{n} u^o(m\Delta t)u^M(\boldsymbol{x}, (n-m+1)\Delta t) \tag{18}$$

なお、区分線形の時間内挿関数は次の通りであり $2\Delta t$ のサポートを持つ。

$$M_T^m(t) = \begin{cases} \dfrac{t}{\Delta t} - m + 1, & (m-1)\Delta t \leq t < m\Delta t, \\ \dfrac{-t}{\Delta t} + m + 1, & m\Delta t \leq t < (m+1)\Delta t, \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

区分線形の内挿関数 $M_T^1(t)$ を点音源からの音として計算された境界値 $u^M(\boldsymbol{x},t), \boldsymbol{x} \in S$, 内点の値 $u^M(\boldsymbol{x},t), \boldsymbol{x} \in D$ も $2\Delta t$ の時間幅のサポートを持つ事から、非ゼロの境界値の記憶に必要なメモリ量や畳み込み演算量を約 $\dfrac{2}{N_T}$ に減らすことができる。

3. 遠方近似を用いた計算

キャラクター位置 $x(t)$ が散乱体から遠く離れているときには、式 (9) 右辺第 2 項、第 3 項に対して遠方近似を用いて計算することにより計算時間の短縮を試みる。遠方近似を用いるにあたり、複数個存在する散乱体の I 番目の散乱体の重心を $\boldsymbol{O}^I$、その境界を S_I $I = 1, 2, \cdots$ とし、$\boldsymbol{x}^I = \boldsymbol{x} - \boldsymbol{O}^I$, $\boldsymbol{y}^I = \boldsymbol{y} - \boldsymbol{O}^I$ とする。式 (9) 右辺第 2 項、第 3 項において、

$$r = |\boldsymbol{x} - \boldsymbol{y}| = |\boldsymbol{x}^I - \boldsymbol{y}^I| \simeq |\boldsymbol{x}^I| - \frac{\boldsymbol{x}^I \cdot \boldsymbol{y}^I}{|\boldsymbol{x}^I|} \tag{20}$$

$$\frac{1}{r} \simeq \frac{1}{|\boldsymbol{x}^I|}\left(1 + \frac{\boldsymbol{x}^I \cdot \boldsymbol{y}^I}{|\boldsymbol{x}^I|^2}\right) \tag{21}$$

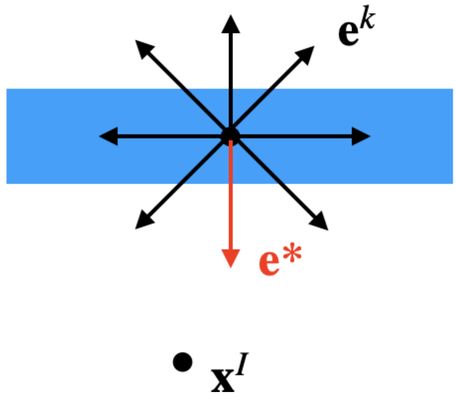

Fig. 3　到達時間差の近似

なる遠方近似を導入し、$\boldsymbol{x}$ と $\boldsymbol{y}$ を分離することで、散乱体 I ごとに $\boldsymbol{y} \in S_I$ に関する積分を予め計算し保持しておく。

また、式 (9) に現れる畳み込み積を行う際には、境界要素 $S_j \subset S_I$ からキャラクター位置 $\boldsymbol{x}$ までの影響の到達時間差を考える必要がある。到達時間差はキャラクター位置 $\boldsymbol{x}$ に依存するが、全ての位置での時間差を予め計算し記憶してくには膨大なメモリ容量が必要となる。そこで、キャラクターの移動をキャラクターが立つ平面である x_1x_2 平面内に制限されることを仮定し、各散乱体上の境界要素からの影響を、x_1x_2 平面を均等に分割した 8 方向

$$\boldsymbol{e}^k = \begin{pmatrix} \cos\dfrac{k\pi}{4} \\ \sin\dfrac{k\pi}{4} \\ 0 \end{pmatrix}, k = 0, 1, \cdots, 7 \tag{22}$$

に限定し、8 方向についての時間差 τ^k

$$\tau^k \simeq n\Delta t - \frac{1}{c}\left(|\boldsymbol{x}^I| - \boldsymbol{e}^k \cdot \boldsymbol{y}^I\right) \tag{23}$$

の境界要素からの影響を事前に計算しメモリにストアしておく。そのうえで、リアルタイムで動くキャラクターと散乱体の方向 $\dfrac{\boldsymbol{x}^I}{|\boldsymbol{x}^I|}$ とが最も近い方向

$$\boldsymbol{e}^* = \left\{\boldsymbol{e}^k \ \middle|\ \max_k\left(\frac{\boldsymbol{x}^I}{|\boldsymbol{x}^I|} \cdot \boldsymbol{e}^k\right)\right\} \tag{24}$$

を求め、ストアされている $\boldsymbol{e}^*$ 方向の到達時間差を用いて影響を計算する (**Fig.3**)。つまり、式 (9) 右辺第 2 項について、キャラクターから遠方の散乱体 S_I からの影響は近似を用いて次のように求める。

$$
\begin{aligned}
&\int_{S_I}\int_0^{n\Delta t}\frac{\delta\left(n\Delta t-s-\dfrac{r}{c}\right)}{4\pi r}\frac{\partial u^M}{\partial n}(\boldsymbol{y},s)dsdS\\
\simeq&\sum_{\{j|S_j\in S_I\}}\frac{1}{4\pi|\boldsymbol{x}^I|}\left(1+\frac{\boldsymbol{x}^I\cdot\boldsymbol{x}^{jI}}{|\boldsymbol{x}^I|^2}\right)\\
&\frac{\partial u^M}{\partial n}\left(\boldsymbol{x}^{jI},\tau^*\right)H\left(n\Delta t-\frac{|\boldsymbol{x}^I|}{c}\right)|S_j|\\
=&\frac{1}{4\pi|\boldsymbol{x}^I|}H\left(n\Delta t-\frac{|\boldsymbol{x}^I|}{c}\right)\left\{\sum_j\frac{\partial u^M}{\partial n}\left(\boldsymbol{x}^{jI},\tau^*\right)|S_j|\right.\\
&\left.+\frac{\boldsymbol{x}^I}{|\boldsymbol{x}^I|^2}\cdot\sum_j\boldsymbol{x}^{jI}\frac{\partial u^M}{\partial n}\left(\boldsymbol{x}^{jI},\tau^*\right)|S_j|\right\}
\end{aligned}
\tag{25}
$$

ここに、

$$
\begin{aligned}
\tau^*&=n\Delta t-\frac{r}{c}\\
&\simeq n\Delta t-\frac{1}{c}\left(|\boldsymbol{x}^I|-\frac{\boldsymbol{x}^I\cdot\boldsymbol{y}^I}{|\boldsymbol{x}^I|}\right)\\
&\simeq n\Delta t-\frac{1}{c}\left(|\boldsymbol{x}^I|-\boldsymbol{e}^*\cdot\boldsymbol{y}^I\right)
\end{aligned}
\tag{26}
$$

であり、$|S_j|$ は境界要素 S_j の面積で境界積分は Gauss の 1 点積分を用いている。式 (9) 右辺第 3 項についても同様の遠方近似を用いて求める。

4. 計算精度について

内点計算の近似の精度検証のため、次の解析解を持つ Neumann 境界値問題を解き、内点における解析解と近似解との誤差を求める。**Fig.4** に示すような中心が $(0.25, 0.25, 0.075)$m で、x_1, x_2, x_3 軸方向に平行で、辺の長さがそれぞれ 0.5, 0.5, 0.15m の散乱体を持つ 3 次元無限領域を D とし、静止した点音源の座標を $\boldsymbol{x}^{\mathrm{o}}=(0.25, 0.25, -2)$m とする。入射波、Neumann 境界条件は下記の通りである。

$$
u^{\mathrm{o}}(t)=\left(1-\cos\frac{2\pi}{\Lambda}t\right)H(t) \tag{27}
$$

$$
\bar{q}(\boldsymbol{x},t)=\frac{\partial u^{\mathrm{o}}(t-r_o/c)}{\partial n} \tag{28}
$$

ここに、$\Lambda=1.66\times10^{-3}$sec とした。境界 S を 640 個の三角形要素に分割し、$\Delta t=1.66\times10^{-4}$sec、波速は $c=340.29$m/sec とする。遠方近似を用いて、受音点を $\boldsymbol{x}=(0.25, 0.25, 0)$m から x_3 軸負の方向に遠ざけながら内点計算をしたところ、散乱体の中心からの距離と 2 ノルム誤差の関係は **Fig.5** のようになり、散乱体の中心からの距離が 2m あたりからは、先行研究 [9] で示した遠方近似を用いずに内点計算を行った際の数値誤差 (2.4×10^{-2}) と同程度となり、許容できる誤差となることがわかった。

5. Unity を用いたゲームアプリへの実装

バーチャル空間の構築には、Unity Version 2020.1.13f1 Personal を用いる [10]。中心がそれぞれ (0.25, 0.25, 0.075)m,(4.075, 0.25, -2)m で、x_1, x_2, x_3 軸方向に平行で、各辺の長さが 0.5, 0.5, 0.15m の 2 つの散乱体を持つ 3 次元無限領域を D とし、静止した点音源の座標を $\boldsymbol{x}^{\mathrm{o}}=(0.25, 0.25, -2)$m とする。

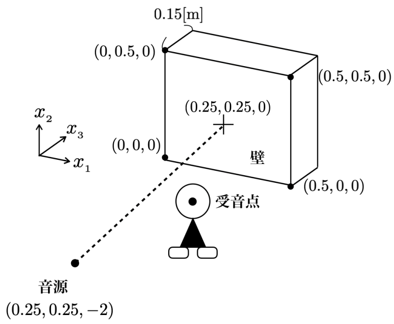

Fig. 4　解析領域

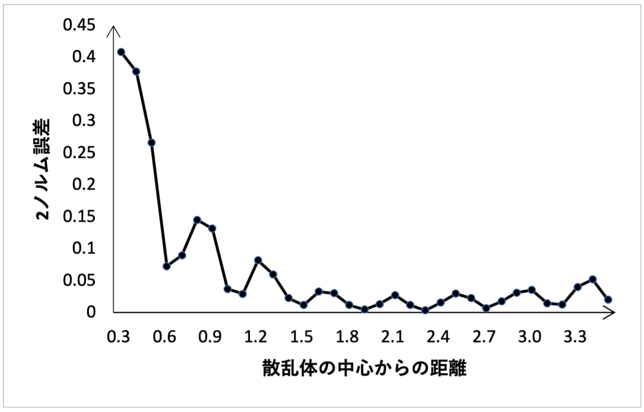

Fig. 5　散乱体の中心からの距離と 2 ノルム誤差の関係

点音源から発信する音 $u^{\mathrm{o}}(t)$ を式 (27) と同じとする。ここに、$\Lambda=1.33\times10^{-3}$sec とする。また、境界条件は境界 ∂D 全てにおいて斉次 Neumann 条件とした。時間域境界積分方程式法による数値解析では、1 つの散乱体表面を 640 個の三角形要素で分割する (計 1280 要素)。また、サンプリング周波数を 12kHz($\Delta t=8.33\times10^{-5}$sec) とする。フレームレートは 30fps を用いる。Unity はサンプリング周波数が 48kHz であり、音声を出力する関数 [11] は $\Delta\tau=\dfrac{1028}{48000}=0.021$sec 毎に呼び出されるため、0.021sec ごとにキャラクターの位置情報の取得更新を行い、移動する内点の音場計算を行った。キャラクター位置と散乱体の中心との距離が 2m 以下の場合には内点計算式 (9) により計算し、2m を超える場合には遠方近似を用いた内点計算により計算する。48kHz の音の作成に必要となるデータ数は 0.021sec 毎に 1024 個であるが、本研究では 256 個のデータを計算し、線形補間により 1024 個分のデータを作成する。この操作を繰り返すことで、移動するキャラクターが聞く音を計算し出力する。

開発した Mac 用のアプリを下記 URL に掲載する。

`http://www-cm.acs.i.kyoto-u.ac.jp/yskw/SoundVR_2021jascome.app.zip`

6. 結論

標準的なスペックのラップトップ PC(MacBook Pro 2.2 GHz クアッドコア Intel Core i7) でアプリを起動し、2 つの散乱体を持つ空間内 (**Fig.6**) でキャラクターを動かしたところ、

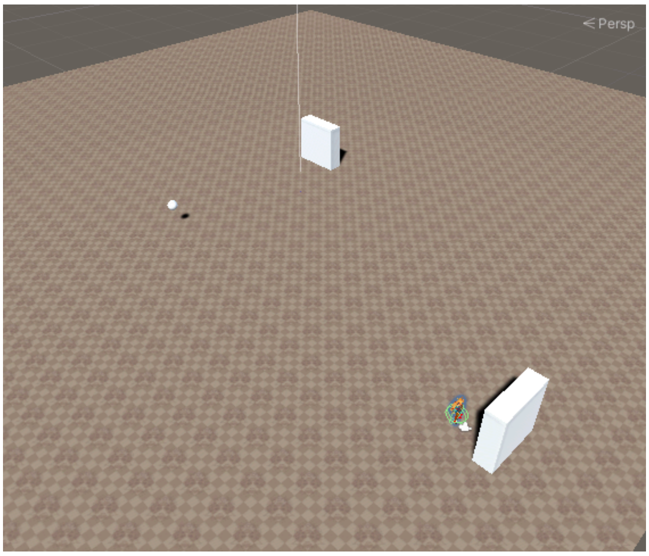

Fig. 6　2 つの散乱体を持つ仮想空間

キャラクターが聞く音をリアルタイムで再現できていることが確認できた。キャラクターを散乱体の近くを移動させることで、2 つの散乱体からの散乱波の影響も表現できていることが確認できた。遠方近似を用いた計算ではリアルタイムに実行する計算量が少なく、散乱体が 2 つ以上の場合にも十分リアルタイム可聴化が可能であると考えられる。

今回行った実装での音の作成方法では、音の作成間隔である 0.021sec 間の音同士のつなぎ合わせが連続的でないことにより、ノイズを含んだ音が鳴ってしまっている。音を連続的につなげることでノイズを含まない音を作成することは今後の課題である。また、今回の解析では音源を静止している 1 つの点音源とした。解いている問題は線形の問題であるため、音源が複数ある場合は同様の数値解析の結果を重ね合わすことで対応可能である。しかし、移動する音源を扱う場合には、移動する音源に対する応答は事前計算ができないため、今回の手法では対応が困難である。幾何音響理論に基づいた手法と提案手法をうまく使い分けるなど、移動音源への対応も今後の課題である。

謝辞

Unity でのゲームアプリ制作に協力頂いた川口功氏に厚く御礼を申し上げ、感謝の意を表します。

参考文献

(1) アスキーストア HP
`https://ascii.jp/elem/000/004/063/4063805/`

(2) 岸智也, 小島健二, 中原雅考, 羽入敏樹, 星和磨: テレビゲームにおけるサウンドエフェクト:インタラクティブリバーブの開発 (小特集:音楽制作を彩る音づくりの技術‘ エフェクタ”), 日本音響学会誌, **68-7**(2012), pp.362-368.

(3) 田近伸二, 樫山和男, 志村正幸: VR 技術を用いた対話型道路交通騒音評価システムの構築, 応用力学論文集, 土木学会, **13**(2010), pp.231–240.

(4) 谷川将規, 守屋陽平, 江嶋孝, 樫山和男, 志村正幸: VR 技術を利用した道路交通騒音評価システムの立体音響化と現実感向上に関する研究, 土木学会論文集 A2 (応用力学), **69-2**(2013), pp.I_155–I_162.

(5) Raghuvanshi, N., Snyder, J., Mehra, R., Lin, M., Govindaraju, N.: Precomputed Wave Simulation for Real-Time Sound Propagation of Dynamic Sources in Complex Scenes, ACM Transactions on Graphics, **29-4-68**(2010), pp.1–11.

(6) 尾本 章, 柏 紘: VR における音場再生の役割, 日本バーチャルリアリティ学会誌, **25-2**(2020), pp.13–18.

(7) 小林昭一他: 波動解析と境界要素法, (2000), 京都大学学術出版会.

(8) 石床竜一, 吉川仁: 時間域 BIEM による音場解析結果の VR 空間での可聴化について, 計算数理工学論文集, **18**(2018), pp.81–84.

(9) 吉川仁, 鈴木賢人: 時間域 BIEM を用いたゲームエンジンによる 3 次元空間を移動する受音点のリアルタイム可聴化, 計算数理工学論文集, **20**(2020), pp.89–94.

(10) Unity のホームページ
`https://unity3d.com/jp`

(11) KAYAC engineers’ blog
`https://techblog.kayac.com/dynamic-waveform-generation-without-audio-clip`