第３部　オンライン講座の運用改善技術報告（その１）

「音声の仮想教室マッピング」開発可能性に関する調査報告

目次

１．音声の仮想教室マッピングの考え方

ドリル訓練アプリでは、世界中に生徒が散在していて、インターネット回線でつながれている。通常の教室とは異なり教師側から生徒の位置関係を把握することは著しく困難であり、参加する生徒も同一空間で学習しているとの臨場感からはかけ離れた感覚に置き去りにされていることになる。平成 30 年度当初においては、どの遠隔教育システムも生徒を仮想空間に配置してその仮想的な位置を聞き分けて生徒を個別に識別する機能を有していない。

ここで、生徒を仮想的教室空間の座席に仮想的に着席させてその位置からの発話であるかのような仮想現実空間を実現すれば教師の生徒把握の負荷も減り、参加する生徒の臨場感も高まる。

仮想教室に生徒を仮想的に座らせるためには、

１．発話者ごとの音源をモノラル音源からステレオ音源を再構成する。

２．奥行きを実現するために後ろの座席からの発話には「音声の奥行」を持たせる。

ここで、通常の意味でのステレオ技術は、基本的に左右の位置の違いを仮想的に実現するだけで、奥行きを仮想実現することはないことを述べておく。奥行きを仮想実現するためには、別の技術が必要である。

なお、ステレオ技術においてはマイクロ秒台の音声の時間遅延調整が必須だが、通常のステレオ音源装置は高額なもので、生徒が各自これを買うことはできない。また、購入したとしてもリアルな左右位置関係を再現するのみなので、大陸をまたがって点在して現に教室にいない生徒たちの位置関係をまるで教室内にいるかのように捉えることはできない。

また、通常のアプリのように高級言語でマイクロ秒台の音声の時間遅延調整を行うことはほとんど不可能である。高級言語が稼働する環境には様々なオーバーヘッドがあり、マイクロ秒程度の遅延は予定外に常時発生していると考えられるためである。これを回避するためには、OS に近い部分でのツール群の探索（発見）または新規制作が想定されるところであった。当チームは、探索の結果、Google 社が提供する「Resonance Audio（立体音響ツール)」が、この目的のツールとして使用できることを突き止め、このツールを使用すれば「音声の仮想教室マッピング」を実現できることが十分推測できる調査結果を得ることができた。

２．前年度の成果との関係

前年度（平成 29 年度)、当チームはドリル訓練アプリの試作に取り組んでこれを完成させている（「ドリル訓練アプリ」の開発」（平成 30 年 3 月 12 日報告）参照)。試作されたドリル訓練アプリを発展させ効果を飛躍的に向上させると期待される「音声の仮想教室マッピング」の技術的実現可能性を調査し検討した。

昨年度実施された「ドリル訓練アプリ」の開発」においては、次の 3 点が重点的に調査された。

１．教師生徒の個別発話の映像と音声の分離保存と分離再生
　結果として映像と音声が発話者ごとに分離できることが確認できた。

２．遅延 0.1 秒程度以内での応答性
　0.1 秒程度以内での応答性は実現できなかったが、パソコンを用いた実機実験で 0.1 秒台の応答は実現できた。実機テストの実感としてはストレスを感じないものであった。

３．発話者の仮想的空間マッピング
　前年度はその必要性が認識されながら、予算の都合上詳細な検討と試作を断念したものであり、本年度、改めて「音声の仮想教室マッピング」として詳細な調査検討を行った。

以下に、それぞれについて「音声の仮想教室マッピング」との関連性を説明する。

２．１　映像と音声の発話者ごとの分離

平成 29 年度においては、発話者の映像と音声が分離できるテレビ会議システム等遠隔教育向けの情報システムは存在せず、これらのシステムの利用者はモノラル音源に頼らざるを得なかった。特定の話者、例えば教師の声だけを消して聞く、またはターゲットされた生徒だけの声を聴くなどのことは不可能であった。しかし、昨年度の当チームの試作システムにおいては個別録音・個別録画ができることを証明することができた。

一方、今後開発されるべき「音声の仮想教室マッピング」システムは、生徒一人一人の発言を、個別に仮想教室の中での正しい位置での発言のようにステレオ化や奥行き化するように個別に加工されなければならない。正しい加工できるかどうかは別途本報告書で検討結果を述べるが、加工の前提となるのが、この「映像と音声の発話者ごとの分離」である。

２．２　遅延 0.1 秒台での応答性

0.1 秒程度以内での応答性は実現できなかったが、パソコンを用いた実機実験で 0.2 秒台の応答は実現できた。実機テストの実感としてはストレスを感じないものであった。

通信速度は、パソコンやサーバのクロック数によって左右される。クロック数の上昇には近年若干の鈍化がみられるが、３年で約 3 倍程度にはなっている。

なお、遅延時間は、カメラにおける AD 変換、カメラからパソコンへの伝送時間、直前データとの差分抽出、ネットへのバケット送出、相手先パソコンがネットからのパケット受信、グラフィックメモリへの

画素展開等の総和であるが、関係するマシンのクロック数が上がれば、それに応じて通信速度が向上し、遅延時間は短くなる。３年すれば遅延時間は 0.05 秒程度になることが予想された。

したがって、ドリル訓練用アプリの通信速度は、パソコンベースで WebRTC を用いることを前提に、通信環境の経年進化改善にゆだねることにした。

２．３　発話者の仮想的空間マッピング

ドリル訓練では、「発話者の仮想的空間マッピング」が必要になる。

たとえば、リアルな語学の授業では、一人の教師が多数の生徒に一斉に発声を促して、その反応を教師が聞く場合、ベテランの教師ならば一人一人の発声を聞き分け、たとえば「田中さん」の発音であるか、せめて「右 2 列目の前から 4 番目あたり」の人の発音であるかを聞き分けて、その人またはその一人周辺を指して、良い発音をほめたり、誤った発音を糺したりすることができる。

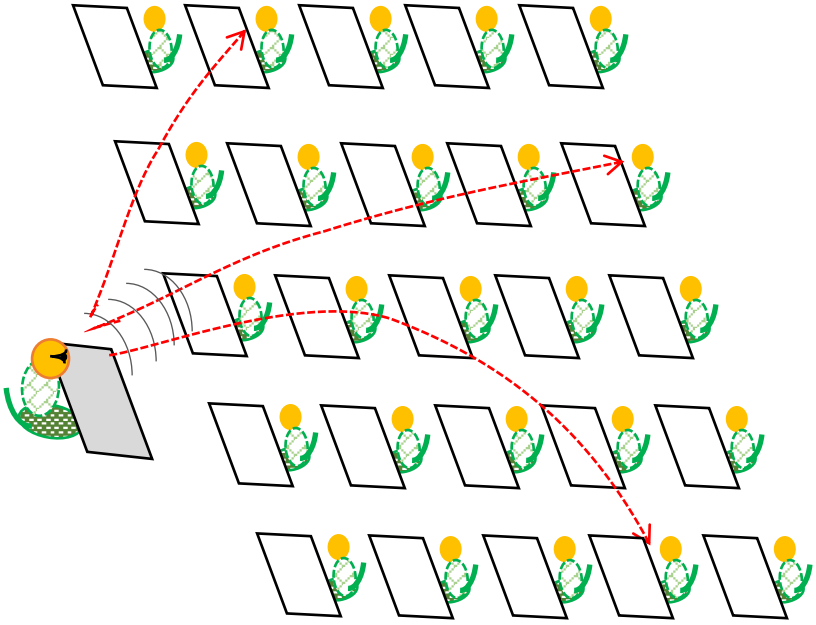
つまり、教師は、生徒の発話の音質や癖ともに空間位置を頼りにどの生徒の発話であるかを知って、ほめたり注意したりすることができるということである。

言いかえれば、生徒の特定のために教師が瞬間的に利用しているものは次の通りである。

・発話した生徒の声の音質や癖

・発話した生徒の声の空間的位置

通常のテレビ会議システムでは、生徒の音声がモノラル・ミックス状態で教師



の耳に達するので、切り取[illegible]顔[illegible]や[illegible]癖[illegible]顔[illegible]生徒を区別しなければならないが、教師には極めて大きなストレスで、しかも判別ができないことのほうが多い。人は、だれの発話かを知るために、音源の位置に多くを依存しているのである。

「発話した生徒の声の音質や癖」「発言した参加者の声の音質や癖」すなわち「発話者の音質や癖」は通信品質のよいテレビ会議システムであれば保証されるが、「発話した生徒の声の空間的位置」や「発言した参加者の声の空間的位置」すなわち「発話者の空間的位置」は現状提供されていない。

ドリル訓練においては、将来複数音源が同時に存在する空間を音源の仮想マッピングに沿って再現す

る技術が必要となる。

音源の仮想マッピングに沿って再現するためには音声の分離保存と分離再生が前提となる。分離再生できなければ音源の仮想マッピングは不可能だからである。

本件について、昨年度は具体的な取り組みを行わなかったが、本年度、今後の開発に向けて調査検討をおこなった。

３．Resonance Audio（立体音響ツール）について

３．１　Google の「Resonance Audio（立体音響ツール)」

ここでは、Google が公開している「Resonance Audio（立体音響ツール)」について、ICT による仮想教室を実際の教室として仮想立体教室化への応用可能を分析し報告する。今回の報告では開発までは行わず、複数の技術仕様書から実現の可能性を探る。

Google の「Resonance Audio（立体音響ツール)」とは、マルチプラットフォーム（パソコンからタブレット PC 及びスマートフォン）の仮想空間オーディオ SDK（ソフトウェア開発キット）であり、仮想空間の音響効果を高い忠実度で提供するとされている。この空間オーディオ技術は、AR(Augmented Reality・拡張現実)、VR（Virtual Reality・仮想現実)、ゲーム、およびビデオにとって非常に有益な技術とされている。

３．２　その他の SDK（ソフトウェア開発キット）

実は Google 以外にも AR（Augmented Reality・拡張現実)、VR（Virtual Reality・仮想現実）を実現する SDK（ソフトウェア開発キット）は増えており、案件毎に開発者が開発するのでは無く、SDK（ソフトウェア開発キット）を利用した開発に徐々に移行はしているが、汎用的なツールは目的の開発物にとって必要にして十分な条件が整わない場合も多く、ICT による仮想教室を実際に実現しようとする場合にも同様の問題が予測される。

さて、昨年度は ICT を利用したリモート教室を実現するために既に Google が開発した WebRTC という WEB ビデオの環境を利用して一部の開発を進めていたので、同一メーカーであるために整合性の高さという点で「Resonance Audio（立体音響ツール)」には大きな期待ができる。日本国内にはこれらを組み合わせた事例も無く、海外のレポート等を参照しながら実現の可能性について探った。

Google 以外にも AR（Augmented Reality・拡張現実)、VR（Virtual Reality・仮想現実）を実現する SDK（ソフトウェア開発キット）については、WebRTC との整合性に確信が持てないため、今回は調査対象から除外することにした。

３．３　Resonance Audio の基本コンセプト

初めに Google の「Resonance Audio（立体音響ツール)」の基本コンセプトを動作の原理から調べた。「Resonance Audio」は限られた資源（CPU 性能・記憶容量）しか持たないモバイルプラットフォームにとって特に有利な、コスト効率と品質の利点を備えている。したがって、映画制作のような世界が中心だった AR（Augmented Reality・拡張現実)、VR（Virtual Reality・仮想現実）をモバイル機器でも実現するという構想に基づいているものということができる。

４．Resonance Audio の基本動作

Resonance Audio は、実際の音波が人間の耳や環境とどのように相互作用するかを再現しようとして、人間の耳と相互作用する音波をシュミレーションするとされている。現実の世界では、音波の相互作用を使って、音の水平位置とその高さを決定する。Resonance Audio は、これらの相互作用デジタル処理で再現して、仮想世界の特定の場所から来る音の錯覚を作成する。

４．１　両耳間の時間差

左耳と右耳に音波が到着する時間の差は、低周波音の水平位置を決定する要素となる。

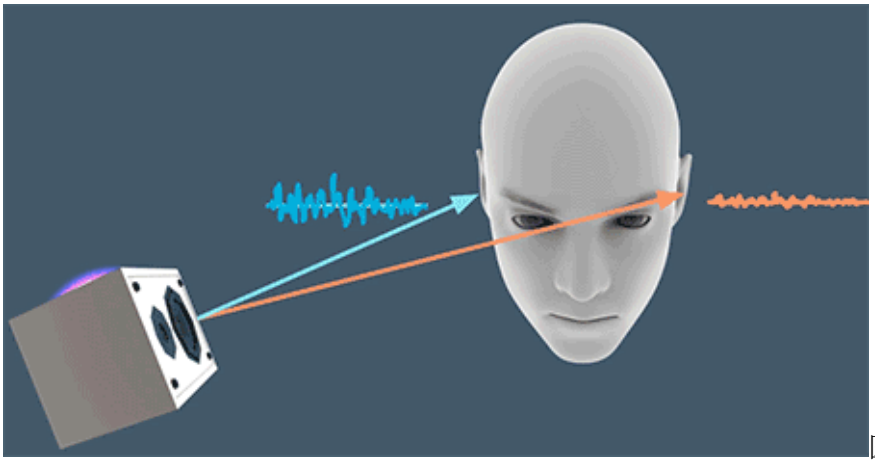


図１

両耳間時間差（ITD,interaural time difference)は音源の水平位置に依存し、どちらかの音源が頭の左か右の先に位置するほど時間差は大きくなる。（図１）

４．２　両耳間のレベル差

人間が高い周波数の音の位置を特定するためには両耳間の時間差（ITD）は影響しないので、代わりに両耳間のレベル差（ILD）を使用して水平位置を決定している。これらは人間の頭のアコースティック・シャドウ（音響陰影・頭の形状により音が光のように遮られる現象）によって引き起こされる左右の耳のラウドネス（人間の耳が周波数によって感度が変わる事）と周波数分布の違いを利用している。

４．３　スペクトル効果

時間とレベルの違いが水平方向に音の位置を特定するが、他の音波の相互作用は音の高さを判断する基準になる。異なる方向から来る音は、異なる方法で私たちの外耳の内側から跳ね返る。（図２）

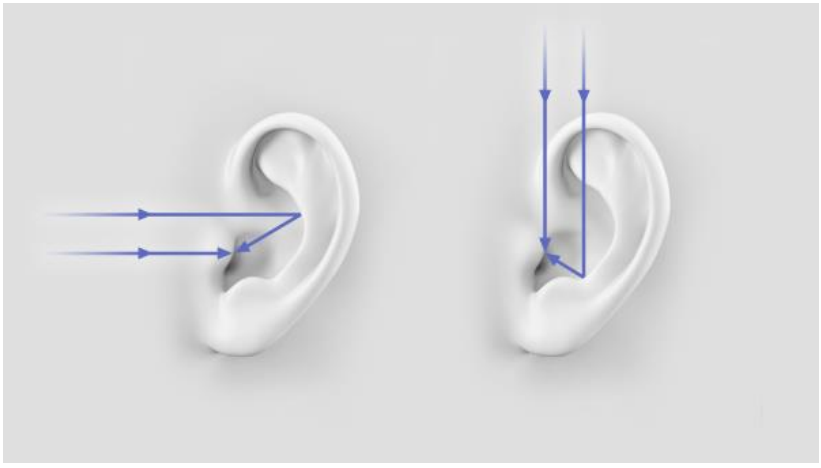


図２

人間はこれらの変化で周波数とスペクトル効果を感じ取り、音源の垂直位置を決定している。

４．４　頭部伝達関数（HRTF）を用いたオーディオキューのシミュレーション

実際の音波と耳との相互作用をシミュレートするために、Resonance Audio は 頭部伝達関数（HRTF, Head-Related Transfer Function）を使用する。HRTF には音の位置を特定するために、音の遅延時間とレベル差とのスペクトル効果を利用する。ヘッドフォンを介しての HRTF で処理された聴覚オーディオ効果は、ユーザに対して周囲の仮想世界内の特定の位置に音源があるという錯覚をもたらす。

４．５　環境と相互作用する音波のシミュレーション

Resonance Audio は、耳と音波の相互作用をシミュレートすることに加えて、環境との音波の相互作用もシミュレートする。

４．６　頭の動きとサウンドの位置

頭を動かすことでオーディオの相対的な変化を知ることができる。Resonance Audio は、これらの頭の動きに反応し、音源の位置を特定の音響空間内に維持する。（図３）


図３

頭部に装着されたディスプレイにより、ユーザの頭部の動きは追跡される。Resonance Audio はこの情報を使用して、ユーザの頭の動きに対応して音響空間を移動させる。以上により、仮想音はユーザに対してその相対位置を維持する。

４．７　初期反射とリバーブ

現実の世界では音波が空気中を移動するにつれて、音波は私たちの環境のあらゆる場面で反射を繰り返す。これにより反射は複雑な組み合わせになる。Resonance Audio は、このような音波の混合を次の 3 つのコンポーネントに分割する。

「直接音」

私たちの耳に当たる最初の波は、音源から直接私たちに伝わる直接音である。音源の距離が増すとエネルギーが減少する。これは私たちから離れた音が私たちの近くの音よりも音量が小さい理由である。（図４）

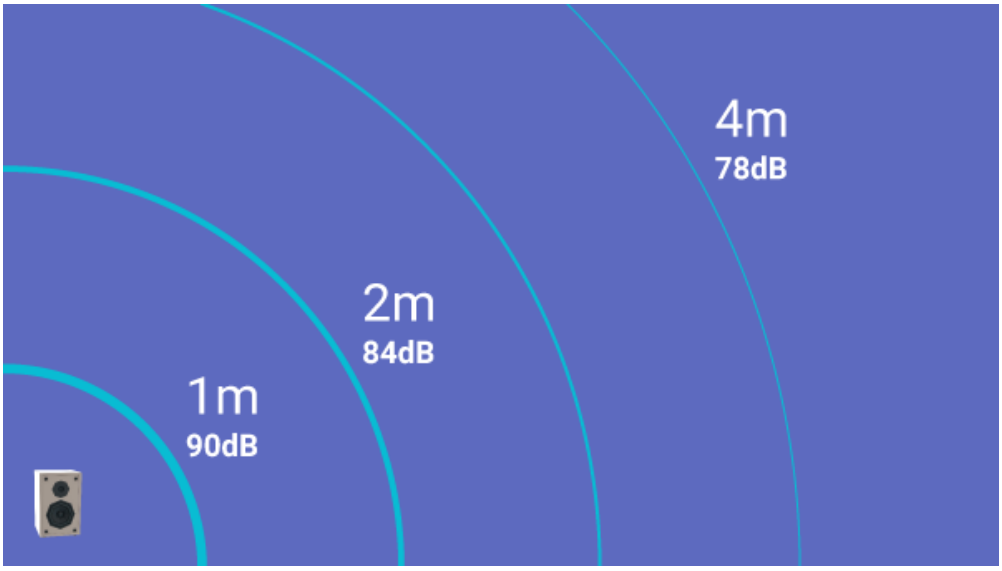


図 4

「初期の反射」

私たちの耳に届く最初の数回の反射波は初期反射と呼ばれている。これらの反射は、我々が位置している部屋の大きさと形状の認識となる。（図５）

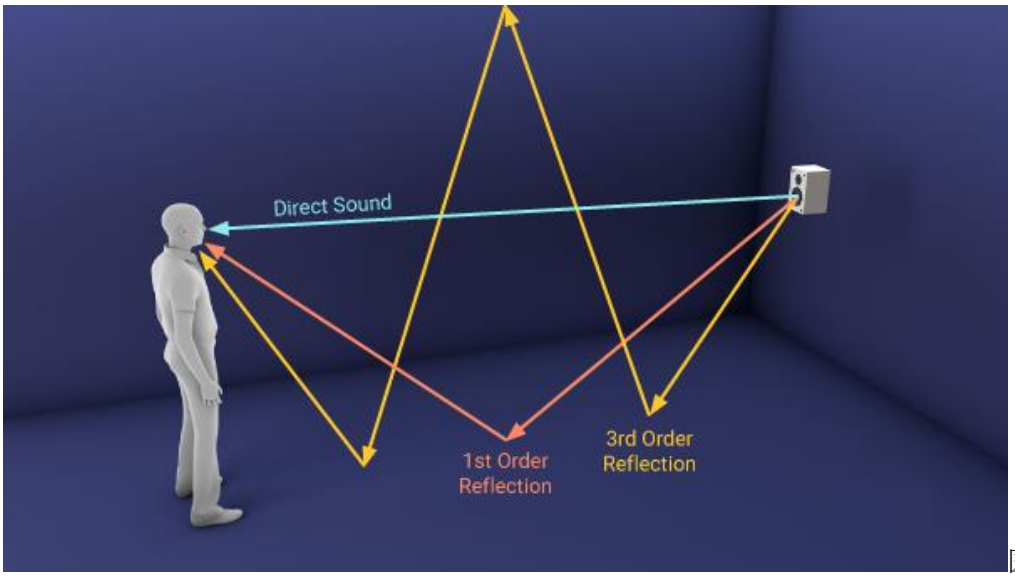


図５

Resonance Audio は、早期反射をリアルタイムで空間化し、各反射のシミュレートされた音源をレンダリング（生成）する。

「後期リバーブ」

時間が経つと、個々の音波が区別できなくなるまで、耳に届く反射の密度がますます高くなる。この現象を後期リバーブという。Resonance Audio には、実際の部屋の音を再現する強力なビルトインリバーブエンジンがある。部屋のサイズや壁の表面材質を変更すると、リバーブエンジンはリアルタイムで反応し、新しい条件に合うように音波を調整する。

「閉塞」

さらなるリアリズムを加えるために、Resonance Audio は、音源とユーザ間を移動する実際の音波がそれらの間のオブジェクトによってどのようにブロックされるかをシミュレートすることもできる。Resonance Audio は、高域成分と低域成分を別々に扱うことによって、これらの環境閉塞効果をシミュレートする。高周波は低周波以上にブロックされて、現実世界で起こることを模倣し再現する。（図６）

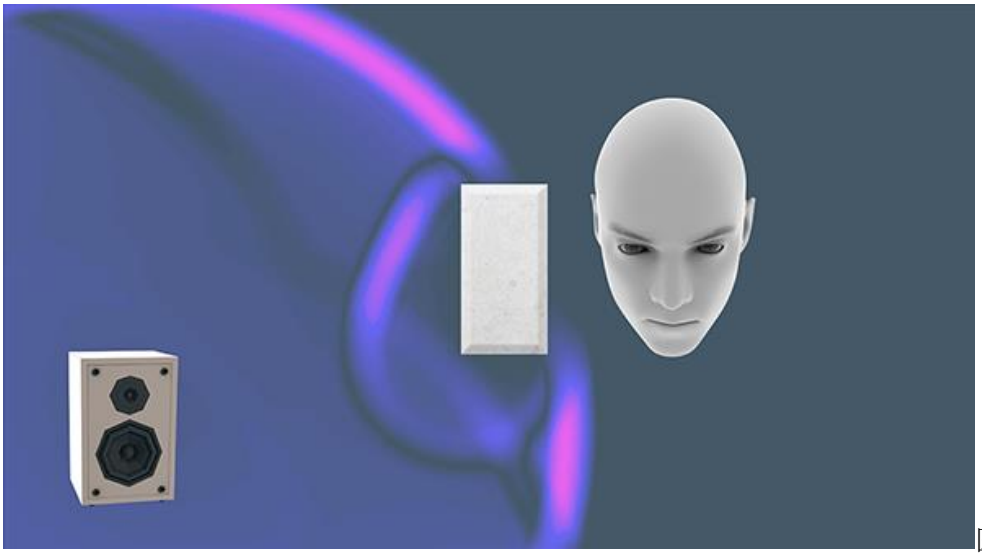


図６

「指向性」

音源の指向性パターンは、オクルージョン（隙間）と密接に関連している。指向性のパターンと形状は、音源から異なる方向に音が出る状況を表現する。音源の音の方向性パターンと音源に対する相対的な位置に応じて、音源の音と異なって聞こえる。一例として、ギターを弾く人の周りを歩いたとする。ギターは、その弦と響き穴がある正面では大きく聞こえ、ギターの後ろに移動するとギターとプレーヤーにより、弦から来る音は遮断される。

（図７）

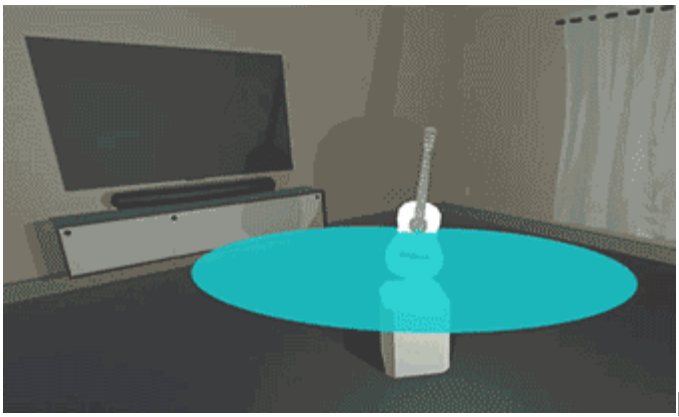


図７

Resonance Audio を使用すると、音源の指向性パターンを変更して実際の音源が音を出す自然で不均一な状況を模倣することができる。

以下の２種類の指向性パラメータが用意されている。

- アルファ：パターンの形状である。円盤、円形、または８つの図形を使用できる。
- シャープネス：パターン幅を表す。

「Ambisonics・再生音に方向感を付ける技術」

Resonance Audio は Ambisonics と呼ばれる技術により、ユーザを音響空間で包み込む。

 図８

アンビソニック指数が増加するにつれて、音波シミュレーションはより正確になる。

５．Resonance Audio の特徴

ここでは、開発する立場として Resonance Audio が持つパフォーマンスとマルチプラットフォームのサポートに最適化された音響空間オーディオ技術についてレポートする。

５．１　オーディオエンジニアリング機能

Resonance Audio は、基本的な 3D 空間と、複雑な音響環境を正確にモデリングする強力なツールが提供される。

５．２　Resonance Audio の能力

- 音源の指向性のカスタマイズ
- ニアフィールド効果（ユーザの耳の近辺の音源）
- 音源の広がり
- ジオメトリベースのリバーブ（立体空間での残響）
- オクルージョン（上下左右の音源に対してこれを隠す物体の効果）
- Ambisonic（音場全体の物理情報）オーディオファイルの録音
- クロスプラットフォームサポート（PC、タブレット、スマートフォン等）

Resonance Audio SDK は、普及しているゲームエンジン（SDK 化されている）、オーディオエンジン、デジタルオーディオワークステーション（DAW）とシームレスに統合され、より実在感のある音響効果を作成することを可能とする。

５．３　コスト

Resonance Audio は、オープンソースでありながらモバイルとデスクトップで利用できるリアルな VR、AR 体験を構築できるので、忠実度の高い大規模な空間オーディオが実現する。

５．４　パフォーマンス

Resonance Audio は、すべての音源を内部的に一括処理して、高度に汎用的な Ambisonic 音場に処理結果を投影する事で、頭部伝達関数（HRTF）をその中の個々の音源への適用では無く、音場に一度だけ設定する事で実現できる機能を提供する。この最適化により音源ごとの CPU パワーの消費を最小限に抑え、従来からの音源別の空間的な手法よりも多くの同時ソースの再生を可能にしている。この解説からも分かるように、複数の生徒を同時に対象とする立体音響空間の ICT 教室構築（仮想教室マッピング）の実現可能性が高い。

５．５　品質

Resonance Audio の Ambisonic 処理は調整可能で、空間分解能が制御できる。Ambisonics を高次元に指定すると忠実度の高い出力が得られ、直接音源の最適化を向上させる。Google によれば、Resonance

Audio のデジタル信号処理アルゴリズムはモバイル環境でも、オーディオ品質を損なうことなく、何百にも及ぶ 3D サウンドソースを同時に音響空間化するように最適化されている。

５．６　Resonance Audio 利用の手順

最初にプラットフォームを構築する。ここではインターネット上で使用する事を前提に話を進めるので、WEB をプラットフォームに指定して、以下の順序で実行する。

- Resonance Audio ソフトウェアをインストールしてプロジェクトに追加する
- 微調整構成の開発者ガイダンスを参照する
- プラグイン用のゲームエンジンの統合（最低一つのゲームエンジンと統合する）

以下に Resonance Audio の資料をご紹介する。この資料は実際に Resonance Audio で開発を進める場合の開発者の基本的な考え方を示している。

「音源」

Resonance Audio は、仮想音源と Ambisonic 音場の空間化を可能にする。サウンドソース（ポイントソースとも呼ばれることが多い）は、仮想環境内の所定のポイントでオーディオソースを定義する。各音源はモノラルオーディオストリームからレンダリングされ、指向性パターンと距離減衰カーブで構成できる。

例えば鳥を音源として用意すると、鳥の羽の音が聞こえたり、近くを飛んだ

図 9

りその場を離れたりすると自然に変化する。

こうした環境全体に音源を配備して、一般的な雰囲気を作り出すことができる。重要なのは音源を作成するときは、余計な音が付加されず、リバーブを含まないモノラルのサウンドファイルを使用する必要がある。

「Ambisonic の音場」

Ambisonics の音場は、視覚的なレンダリングのための光プローブと同様、聴取者の周囲の仮想球面上に音波をエンコードすることによって 360° の空間的な完全な音響空間を表現する。Resonance Audio は 3 次元までの Ambisonics のデコードをサポートし、各 Ambisonic 音響空間は、マルチチャンネル入力オーディオストリームからレンダリングされ、ストリームは常にリスナーの位置を想定してレンダリングされる。

Ambisonic ファイルは頭の回転にのみ反応し、その音場は遠くの音の表現に適する。

図 10

「オーディオルーム」

作成する環境に応じて、オーディオルームを使用する必要がある。インストールされた Resonance Audio には音響環境のモデルとしてオーディオルームが提供され、基本の反射とリバーブを提供している。音響空間内のユーザの位置の近くに壁や構造物がある場合に、音響効果をより現実的にするのに役立つ。リバーブと部屋の構造物のサーフェスを形成している環境の音響効果に合わせて設定を行うモデルである。

オーディオルームは、対象とされる環境が室内である場合に最適であるが、屋外の場面ではオーディオルームを指定すると自然さが低下する。屋外では地面が音響環境に於ける唯一の反射面になる可能性が高いからというのが理由である。

図 11

「サウンドデザインのヒント」

音源をアニメーション化する

ユーザがビューの外に音を鳴らすようにするには、サウンドの位置をアニメート化（動かす事）できる。これにより、ユーザはサウンドの位置を特定できるようになる。(図 12)

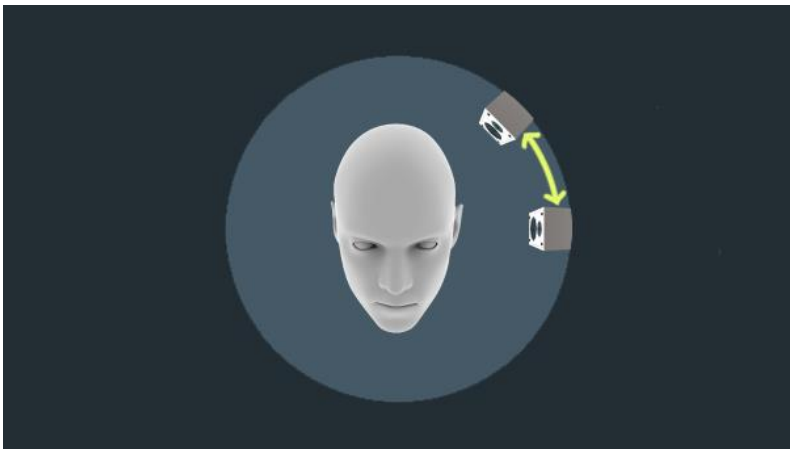


図 12

「サウンドを繰り返す」

ユーザがサウンドの位置を特定しやすくするために、複数回繰り返して再生する。繰り返される音により、ユーザによる認識が容易になる。電話を複数回鳴らすと、着信に気づきやすくなり、電話への応答がスムーズになる。多くの異なる要素からなるサウンドを使用することで、同じ効果を実現できる。

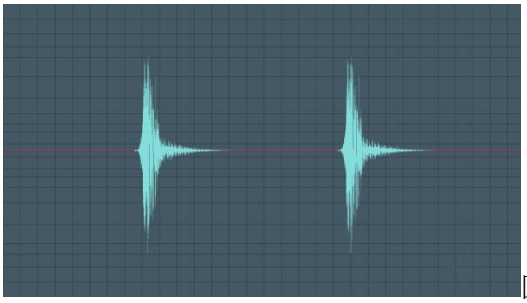


図 13

「複雑なサウンドを使用する」

次のような特性を持った音の使用が推奨されている。

- 十分な音量レベル
- 周波数の全スペクトル
- 複雑

あまりにも静かで、高い周波数を含まず、またはサイン波のビープ音などの単純すぎるサウンドの使用を避けた方が好ましいとされている。

「Ambisonic ファイルの作成」

デジタル･オーディオ･ワークステーション（DAW）ソフトウェアや、Ambix などのプラグイン、Ambisonic ファイルに対する 2 種類の方法で作成することができる。

１．モノフォニックファイルを使用して、ユーザの周囲の仮想空間に音を置く。そうすることによってサウンドを移動してエフェクトを追加することができる。

（図 14）は Digital audio workstation（デジタル・オーディオ・ワークステーション・プラグインによる Ambisonic ファイル作成のサンプル画面）である。

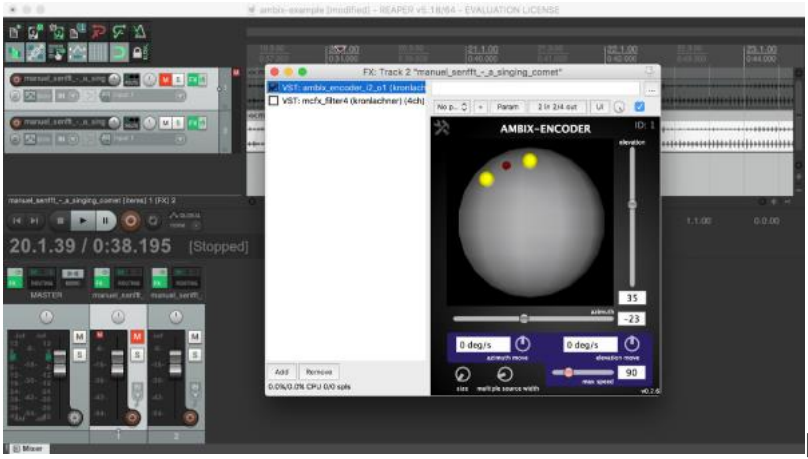


図 14

２．SoundField ST450、TetraMic、または Zoom H2n のような Ambisonic マイクロフォンを使用して、3D の環境音をキャプチャする。キャプチャしたサウンドを Ambix プラグインに読み込んでエフェクトを実行し、必要に応じて回転させてからミックスすることができる。

「ユーザーエクスペリエンスと品質のコントロール」

視覚とオーディオの体験を同期させるにはその場面内でユーザが見ているものが、聞いたものと一致しているかを検証する必要がある。例えば海岸で波がぶつかるのを聞いても、海は動きのないように見える場合、現実的ではないように感じてしまう。

「開発内容を検証する」

開発プロセス全体を通しての作業を検証して、音源と構成が最適なオーディオ体験を提供していることを確認する。

「環境全体のサウンドをテストする」

ユーザが認識できるすべての環境を確認して、音響が自然に響いている事を確認する。通常ユーザが環境内を自由に移動できるような場合は音源を移動する。

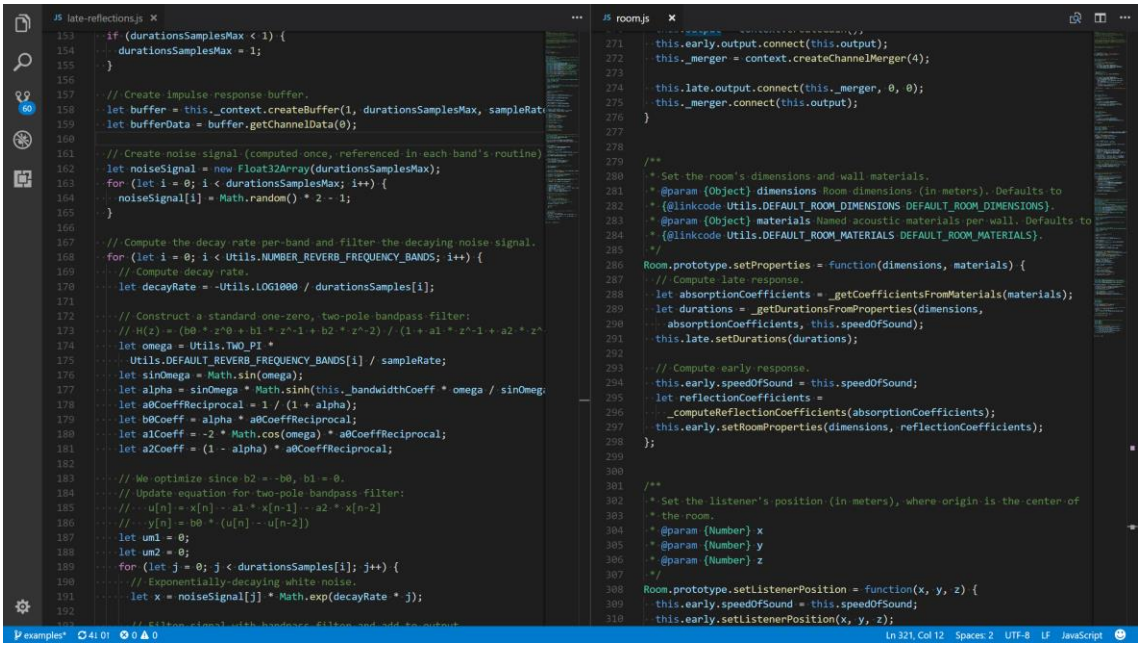
「音質、音量、応答性を確保する」

作成されたサウンドが高品質で、クリアで快適な音量であることを確認し、現実的な動きに合わせて調整する方がよい。

「ヘッドフォンを使ってテストする」
空間オーディオを十分に体験するには、ヘッドフォンが最適である。サウンドのテストはヘッドフォンで行う必要がある。テストのためにはコンピュータ等の一般的スピーカーを使用しない事が重要である。

６．Web 用 Resonance Audio SDK の利用例

（以下、コードサンプル）



Resonance Audio は JavaScript SDK で、Web オーディオアプリケーション用の拡張可能な Ambisonic サウンドフィールドに空間オーディオを動的にエンコードする事でリアルタイムに応答する開発環境と記述されている。JavaScript SDK である事は ICT による授業のシステムを開発した WebRTC との親和性が高い事を意味するので、非常に期待される。ここからは Resonance Audio SDK をインストールして、それを使って音響空間の例を作成する方法をレポートする。以下の手順で開発環境が完成する。

「ウェブプロジェクトを設定する」

Resonance Audio SDK をインストールし、プロジェクトに組み込む。

６．１　SDK のインストール

npmWeb プロジェクトに SDK をインストールするために使用する。

```
npm install resonance-audio
```

６．２　SDK のプロジェクトへの組込み

次のいずれかのオプションを使用して、HTML ファイルに SDK ファイルを含める。

- Resonance Audio のソースコードを変更する予定がない場合は外部サイトより SDK をロードする。

```
<script src = "https://cdn.jsdelivr.net/npm/resonance-audio/build/resonance-audio.min.js"> </ script>
```

- Resonance Audio のソースコードを拡張することを計画している場合、開発者はインストールを使用して組み込む。

```
<script src = "node_modules / resonance-audio / build / resonance-audio.min.js"> </ script>
```

６．３　サンプルの場面の構築

次の手順では、オーディオ出力のサンプルの場面を作成する方法を示す。
場面の空間オーディオを体験するには、ヘッドフォン装着を指定する。

「Resonance Audio の場面を作成する」

① AudioContext と共鳴オーディオの場面を作成する。

```
// Create an AudioContext
let audioContext = new AudioContext();

// Create a (first-order Ambisonic) Resonance Audio scene and pass it
// the AudioContext.
let resonanceAudioScene = new ResonanceAudio(audioContext);
```

② オーディオ出力を設定する。

```
// Connect the scene' s binaural output to stereo out.
resonanceAudioScene.output.connect(audioContext.destination);
```

「音響の場面に部屋を追加する」
レゾナンスオーディオルームのモデリングでは、現実的なオーディオ空間の反射や音響の場面における残響の作成に使用される。

① まず部屋の寸法をメートルで定義する。

```
// Define room dimensions.
// By default, room dimensions are undefined (0m x 0m x 0m).
let roomDimensions = {
  width: 3.1,
  height: 2.5,
  depth: 3.4,
};
```

② 部屋の 6 つの面（4 つの壁、天井、床）の各部屋の材質を定義する。

```
// Define materials for each of the room' s six surfaces.
// Room materials have different acoustic reflectivity.
let roomMaterials = {
  // Room wall materials
  left: 'brick-bare',
  right: 'curtain-heavy',
  front: 'marble',
  back: 'glass-thin',
  // Room floor
  down: 'grass',
  // Room ceiling
  up: 'transparent',
};
```

③ 部屋の定義を場面に追加する。

```
// Add the room definition to the scene.
resonanceAudioScene.setRoomProperties(roomDimensions, roomMaterials);
```

「場面にオーディオ入力ソースを追加する」

① AudioElement オーディオソースファイルを用意してロードする。

```
// Create an AudioElement.
let audioElement = document.createElement('audio');

// Load an audio file into the AudioElement.
audioElement.src = 'resources/SpeechSample.wav';
```

② AudioElement から MediaElementSource を生成する。

```
// Generate a MediaElementSource from the AudioElement.
let audioElementSource = audioContext.createMediaElementSource(audioElement);
```

③ MediaElementSource を場面に追加する。

```
// Add the MediaElementSource to the scene as an audio input source.
let source = resonanceAudioScene.createSource();
audioElementSource.connect(source.input);
```

「オーディオソースを配置して場面をレンダリングする」

① 場面をバイノーラル（人の頭の耳をマイクとして収録）でレンダリングするには、オーディオソースを部屋の中心（0,0,0）に合わせて配置して再生する。

```
// Set the source position relative to the room center (source default position).
source.setPosition(-0.707, -0.707, 0);

// Play the audio.
audioElement.play();
```

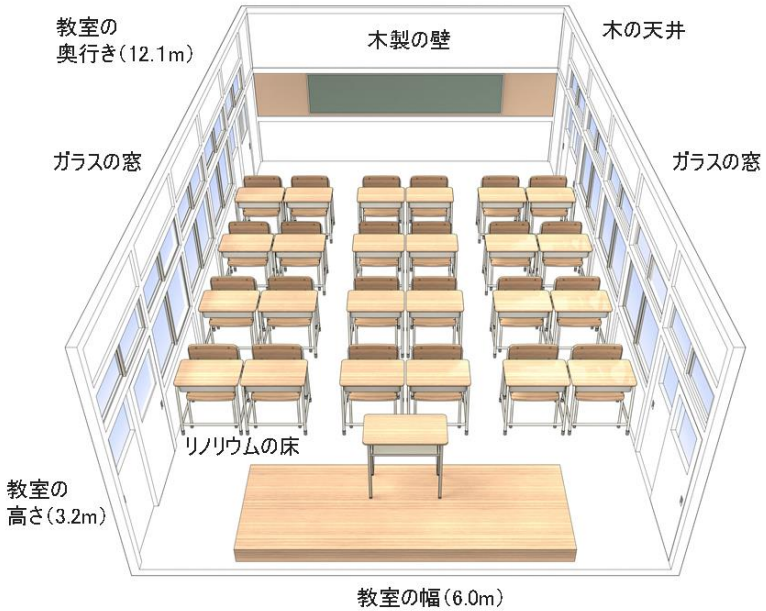


図 15

図 15 のような仮想立体教室を想定して Google の「Resonance Audio（立体音響ツール）」を実現する

コードを以下に記述する。

```
// 最初に教室となる音場を用意します
let audioContext = new AudioContext();
let resonanceAudioScene = new ResonanceAudio(audioContext);

// 音の出力を設定します
resonanceAudioScene.output.connect(audioContext.destination);

// 教室の大きさをメートルで、幅、高さ、奥行きを指定します。
let roomDimensions = {
  width: 6.2,
  height: 3.2,
  depth: 12.1,
};

// 教室の素材を設定します（日本なので木材が多い）
let roomMaterials = {
  // Room wall materials
  left: 'glass-thin',
  right: 'glass-thin',
  front: 'wood-panel',
  back: 'glass-thin',
  // Room floor
  down: 'linoleum-on-concrete',
  // Room ceiling
  up: 'wood-ceiling',
};

//  部屋の定義を音場に追加します
resonanceAudioScene.setRoomProperties(roomDimensions, roomMaterials);

// オーディオ入力ソースを用意して追加します
let audioElement = document.createElement('audio');
audioElement.src = 'resources/SpeechSample.wav';

// 追加されたオーディオ入力ソースからMediaElementSourceを生成します
let audioElementSource = audioContext.createMediaElementSource(audioElement);

// MediaElementSourceを場面に追加します
let source = resonanceAudioScene.createSource();
audioElementSource.connect(source.input);

// 教壇から見て手前の左側の生徒を設定します
source.setPosition(-2.810, -4.620, 0);

// Play the audio.
audioElement.play();
```

７．「Resonance Audio（立体音響ツール)」に関するまとめと考察

Google が公開している「Resonance Audio（立体音響ツール)」について、その全容を追いながら「ICT による仮想教室」への応用を探った。結論から言えば十分に可能性がある事までは理解できた。生徒を音源とした場合、数百までの生徒を音源として三次元の座標で指定して配置して生徒個々を立体的な指向性を持って認識する事が可能となる。既に開発を進めている遠隔授業のシステムはインターネットブラウザを対象としており、またGoogle社のWebRTCのAPIを利用したものであり、開発言語としてJavaScriptを使用する。「Resonance Audio（立体音響ツール)」は複数の開発環境を選ぶ事が可能であるが、WEB を対象とした場合はブラウザに Google 社の Chrome と言語には JavaScript を使用する。ここで重要なのは「WebRTC」も「Resonance Audio（立体音響ツール)・WEB 版」も HTML5 に準拠した技術であり、オーディオ・ビデオ・通信を扱うための技術であるという点である。

「Resonance Audio（立体音響ツール)」と同様に HTML5 に対応する WebGL 等の API も存在する。しかし WebGL（ウェブブラウザで 3 次元コンピュータグラフィックスを表示させるための標準仕様で非営利団体の Khronos Group で管理されている）のドキュメントよると GPU（グラフィックス プロセッシング ユニット、略して GPU）が必要であることがすぐにわかる。教師用のパソコンには GPU も有りえるが、生徒側の様々な端末には GPU には対応できない。「ICT による仮想教室」としての再生は教師側だけで良いとする考えもあるかもしれないが、GPU すら不要な Google の「Resonance Audio（立体音響ツール)」の方がレスポンスに優れるのは当然と推測できる。WebGL も Google の「Resonance Audio（立体音響ツール)」も無償で提供されている API（アプリケーションインターフェース）であり、配布などの問題も無いのも有り難い。

「Resonance Audio（立体音響ツール)」を使わずに開発をした場合、恐らく「仮想教室マッピング」は生徒の位置を認識するための音の遅延と音量の変化程度で終えてしまった事が想像されるが、「Resonance Audio（立体音響ツール)」を採用する事で、以下が実現する。

1. 教室サイズの設定
2. 23 種類に及ぶ壁面や部屋のオブジェの材質まで考慮した音の変化への配慮
3. 実際に音声を聞く教師の耳の位置を考慮したバイノーラルの集音
4. 数百の生徒数を想定

WebRTC に「Resonance Audio（立体音響ツール)」を組み合わせた場合、WebRTC を介しての音声を「仮想立体教室」の各座席に配置するという形の開発になると考えている。各生徒の映像は指定すると別画面での視聴になると思う。

WebRTC と「Resonance Audio（立体音響ツール)」を組み合わせた３D のシステムについては現時点では前例は皆無である。一方、WebRTC 自体を３D 化した音声の伝送に使う試みは海外の WEB サイトに幾つか発見したので、次年度以降はこれらを参考にしながら ICT による「仮想立体教室」の実現に向けて WebRTC と組み合わせた試作とテストを開始したいと思う。

８．仮想教室マッピングシステムの開発に向けて

８．１　「Resonance Audio（立体音響ツール)」教室の仮想立体表現

以下の効果はいずれもResonance Audioでは 頭部伝達関数(HRTF, Head-Related Transfer Function)を使用して実現することができる。次年度、仮想教室マッピングを試作するとすれば、これらの効果が有効に活用され満足度の高いシステムが作れる可能性が高い。

1.水平方向の方位は2つの機能を利用して仮想的に出現させることができる。

・比較的低い周波数領域ではでは左右の遅延時間の差

・高い周波数領域では両耳間のレベル差（ILD）すなわち人間の頭のアコースティック・シャドウ効果に相当する音のレベル（大きさ）の差

仮想教室マッピングにとっては必須の効果である。

2.上下の方向は外耳での音の反射効果すなわちスペクトル効果で仮想的に出現させることができる。教室内は基本的に水平なので上下の方向は不要と思われるが、階段教室を実現する場合や、フラットな教室でも教師だけは一段と高くするなどの工夫をする際は有用な技術となる。

3.奥行については「初期の反射」と「後期リバーブ」の効果を用いて仮想的な奥行きを表現することができる。これも仮想教室マッピングにとっては必須の効果である。

4.障害物についてもブロック効果を用いて仮想的な奥行きを表現することができる。試作して見ないと判断できないが、教室の２列目以降に座った生徒は前に座っている生徒らの体が障害物になっているはずである。教師は前列に座った生徒の身動きなどに伴うざわつき音を雑音としてとらえていることは間違いないが、後ろの生徒の声が前列やその前の列などの生徒の体にさえぎられていることをどこまで耳で感じているがわからない。また生徒の体のモデリングには相応の工数もかかるので、費用対効果を考慮して、障害物についてもブロック効果を仮想教室に実装するかどうかの判断が必要と思われる。

他方、環境との相互作用については、頭部伝達関数（HRTF, Head-Related Transfer Function）の内部で実現するのではなくアプリ側で、音源の位置・速度・加速度などを管理したり、聞き手の頭の位置・速度・加速度などを管理して調整を図ることになる。仮想教室マッピングでは、音源や教師の移動が臨場感を増す場面の出現確率は低いと考えられる。しかし、教師が教室内を循環するような状態をあえて演出するような際には使われる技術である。

８．２　教師の利用体験

様々な先例が示しているようにどんなに優れた仮想現実システムができてもこれを利用する人が事前に使用して慣れていなければ、利用中の不安感やこれに伴う目まいや吐き気など身体症状を起こすことがある。これを完全には防ぐことはできない。

訓練ドリルシステムにこのシステムを採用するにあたっては、講師（インストラクター）に十分な余裕をもって事前体験を繰り返してもらう必要がある。

８．３　通信遅延の問題

左右の方位判別のための遅延時間がマイクロ秒オーダであることと比べて、ネットを介する通信遅延が 0.1 秒台（パソコンの場合）と極めて大きいが、逆に差が桁違いであるだけに弁別領域が重ならないので影響を及ぼさないと考えられる。ただし、教室の前後ろを仮想的に表現する遅延時間とはかなり接近しているので、仮想現実感を妨害する恐れもないとは言えない。すなわち、通信遅延による遅延なのか後ろの席に座ったための遅延なのかが受信者には区別がつきにくくなるという恐れが存在する。この問題の解決のために、現段階では、生徒ごとの通信遅延時間を何らかの方法（例えばスライドバー表示）で示すことを想定している。