第９部

Software Defined Media

SDM WGメンバー

第１章　はじめに

Software Defined Media（SDM）コンソーシアムは，2014年に設立され，オブジェクト指向のディジタルメディアと，インターネット環境が前提の映像・音響空間を用いたビジネス創造を目指してきた．昨年までの報告書では，SDMのコンセプトやアーキテクチャ，試作プラットフォーム[77][78]，データ収録[79]，収録データベース設計[80][81]，SDMプラットホームを利用したアプリケーションであるSDM360^2 [82][83]やLiVRation [84][85]などを報告してきた．

本報告書では，2019年度に実施した活動を報告する．まず初めに，2章では，LiVRationの実用化に関して述べる．次に，3章では，SDMコンソーシアムとしてInterop 2019東京に出展したデモンストレーション概要について述べる．4章では，今年度に新たに設計・実装したWeb360^{2}について報告し，5章では，SDMに関連した卒業論文として発表されるco-Soundについて報告する．6章では，3度目の開催となるSDMシンポジウム2019について報告する．最後に，7章で本報告書の結論を述べ，今後の方向性を議論する．

第２章　LiVRationの実用化

遠隔で音楽ライブをマルチアングル，立体音響をインタラクティブに視聴できるLiVRationを実用化は，NTT西日本，アルファコードが主体となって取り組んだ．現在，REALIVE360を商標として，スマートフォンベースのVRアプリとしてiPhoneとAndroidのアプリケーションとして提供されている（図1，図2）*1．これまでに，アイドルグループのももいろクローバー Zのコンテンツが購入可能である．

以下，NTT西日本と，アルファコードのREALIVE360の事業開始のプレスリリースを挙げる．

- マルチアングルVRサービス「REALIVE360」の有償提供開始について～観たいアングルを自由に選択できるマルチアングルで，まさに未来型ライブ体験～｜NTT西日本*2

図1　REALIVE360の概要（出典ウェブサイトより）

図2　REALIVE360の概要（出典:ウェブサイトより）

*1　https://realive360.jp/
*2　https://www.ntt-west.co.jp/info/support/oshirase20191224.html

• 8K VRを実現する音楽配信サービス「REALIVE360」においてVRライブの配信プラットフォーム・撮影技術提供で共同事業開始｜アルファコード[*3]

第３章　INTEROP東京2019での展示

Internet of Things（IoT）の技術進展によって，視聴空間における視聴者の動きなど様々な要素がリアルタイムに情報空間に取り込まれるようになってきた．さらに，オブジェクトベースの立体音響システムでは3次元の情報空間上の音声オブジェクトをリアルタイムに視聴空間にレンダリングできるようになってきている．

本研究では，IoTを活用することで，インタラクティブな音声視聴を可能とするネットワーク化された立体視聴プラットフォームの開発・評価を行った．また，スマートフォンやタブレット端末などの民生品とAugumentedReality（AR）を活用して，プラットフォーム上でインタラクティブな立体音響操作を実現するアプリケーションを開発した．これらのシステムを，Interop 2019にて展示し，45人の主観評価を実施することで，その有効性を確認した（詳細は[86]を参照）．

システムの構成を図3に示す．図の下部はLANに接続された音響設備・アプリケーションであり，上部はインターネット上に設置されたサービスを示す．本システムでは，アプリケーション間の通信のための通信バスとしてMQTTを採用し，クラウド上にMQTT Brokerとデータ前処理等を行うサービスを配置し，ローカル側に立体音響再生のためのサービスを配置することでプラットフォーム（立体音響プラットフォーム）を構成している．

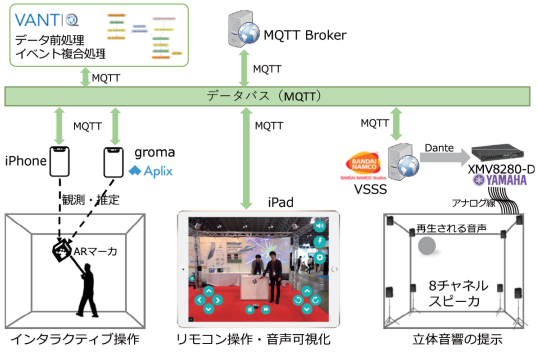

図3　立体音響プラットフォームのシステム構成

試作システムでは，ARマーカによるインタラクティブな音声オブジェクト操作アプリケーション，リモコンUIによる音声可視化アプリケーションの２つがプラットフォーム上に配置されており，これらを組み合わせることで，オブジェクトベースの立体音響を実現している．

本章では，それぞれのサービス，アプリケーションについて述べる．

3.1　インタラクティブ操作アプリケーション

複数のARマーカの位置情報を，iPhoneを使って推定するアプリケーションである．Aplix社が提供しているQuanti™の拡張版を用いており，基準点として設置したARマーカからの位置と方向を推定することができる（図4）．推定されたそれらの情報は，ARマーカの検知IDを含む形でJSONにエンコードされ，MQTTでリアルタイムに発出される．以下はMQTTのペイロードサンプルである．検出したデバイス（device）と，検知したARマーカのID（tagID），姿勢情報（Rx,Ry,Rz），位置情報（t）が発出される．

```
{
    "sequence":45,
    "timestamp":"2019/04/19 10:04:46.3900",
    "device":"iPhone1",
    "tagId":1,
    "withOrigin":true,
    "Rx":{"x":-0.0538,"y":0.0102,"z":-0.9985},
    "Ry":{"x":0.9936,"y":0.0997,"z":-0.0525},
    "Rz":{"x":0.0991,"y":-0.9950,"z":-0.0155},
    "t":{"x":-0.1341,"y":0.1544,"z":0.2120}
}
```

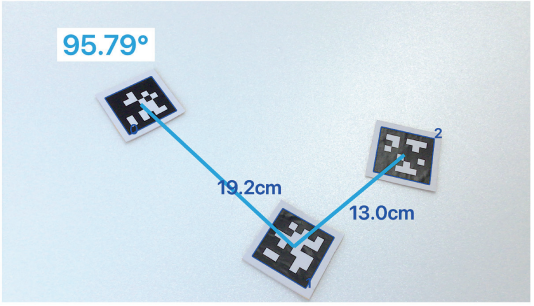

図4　QuantiによるARマーカの検出

*3　https://www.alphacode.co.jp/news-events/news/20191224130846.html

今回実証した環境では，ユーザの動きによってiPhoneの視覚が遮られる場合に備えて，およそ2.5メートルの高さに，iPhoneを2箇所設置した．なお，iPhoneは有線LANに接続し常時給電されている．図5はARマーカを用いた操作の例である．ARマーカを張り付けた指示棒を使って，音声の中心位置を操作している．

後述するVirtual SoundScape System（VSSS）による音声提示が可能なソース同時に 1 つまであるため，複数台のQuantiを用いて立体音響制御を試みた場合，それぞれに推測された位置情報が異なったり，ARマーカをQuantiのフレームレート20fpsよりも高速で動かしたりした場合，再生位置が急に移動したり，ぶれて聞こえたりする懸念があると考えた．そのため，プラットフォーム側の機能として，カルマンフィルタによる位置推定を実装した．

3.2 立体視聴プラットフォーム

プラットフォームの機能としては，以下を実装した．

1. データ（メッセージ）バス
   IoTデバイスやアプリケーション間でのメッセージ交換する機能．

2. データ前処理・イベント複合処理
   特定のメッセージに対して，ルールベース／機械学習による異常値検出や後述するカルマンフィルタによる位置情報の更新などの前処理を行う．

図5 ARマーカによる操作（音声可視化アプリ）

3. データ保存
   アプリケーションの動作に必要なデータをプラットフォーム上に保持する機能．永続化が必要なデータについては，別途データソースを用意する．

4. データ変換
   立体音響の提示のためにデータ変換を行い，MQTTでメッセージを発行する機能．

5. 立体音響の再生
   VSSSを使って，スピーカーシステムから立体音響を再生する機能．

本システムでのメッセージ交換は，以下の流れとなる．Quantiから取得した複数の位置情報は，MQTTで発出された後，プラットフォームで前処理が施される．異常値検出や検知対象の位置情報の推定が行われた後，必要なデータ変換を施されて立体音響の再生を行うスピーカシステムであるVSSSに操作コマンドとして入力される．具体的には，データ変換時の処理として，前処理された位置座標はVSSSが解釈可能な値の幅（0 ～ 255）に変換され，コマンドの一部として入力されている．この際，再生対象の音源選択のコマンドは，リモコン操作部からの入力を，プラットフォーム側が保持し，必要に応じて再生側に伝えることで実現される．これらの操作コマンドは，VSSSの入力インタフェースとして提供されていたMIDIによるAPIをMQTTでラップしたものとなっている．なお，インターネットに設置したMQTT Brokerのサーバスペックは表1である．

MQTTトピックの命名規則としては，表2に示す固定の11階層の構造を採用した．これらは，建物設備制御における命名規則[87]を踏襲しており，建物の空間内に配置された設備機器の情報を十分に記述できることに加え，BIM（Building Information Modeling）との相性も良い．例えば，

表1 MQTT Brokerのサーバスペック

OS	Ubuntu 16.04.6 LTS
CPU	Intel(R) Xeon(R) E5530 @ 2.40GHz
メモリ	8 GB
MQTT Broker	VerneMQ

３次元空間の特定の位置に特定の音を再生する場合，以下のようなTOPIC名でコマンドを発行する．

```
1  3dbcs.biz/UTokyo/iREF/6F/Hilobby/SDM/midi/sc/
2  VSSSO1/controlChange/W
```

3.2.1　データ前処理

データ前処理・保存・変換の一部は，リアルタイムイベント処理基盤であるVANTIQを用いて実装している．機能モジュールを組み合わせることで容易にシステム開発・デプロイが可能であり，かつリアルタイム性・信頼性の高い商用システムであるが，同様の機能は，OSSであるNodeRED[*4]などでも実現できる．図6にVANTIQで開発した前処理のロジック一部を示す．

Sourceと記載のあるノードがMQTTからの入出力を示しており，App，Ruleと記載あるノードで詳細なロジックの記述を行っている．処理の概要としては，実証環境やシミュレータから生成した，位置情報データに対しては前処理を行った上で，カルマンフィルタによる位置情報の推定器(Estimator)に送るために，MQTTの通信バスに発出している．Estimatorからの入力に対しても前処理を行い(App:Preprocess_From_Estimator)，SDM用のデータ変換(Rule:format_TagInfo_For_SDM)を行うことで，立体音響の再生が行われる．なお，それぞれのノードで扱うデータに対しては，Typeノードによってスキーマ定義がされる．

表2　MQTTトピック命名規則

属性	階層	説明
Who	1	サービス事業者名
Where	2	敷地名・サイト名
	3	棟・建物名
	4	階数
	5	分割エリア・フロア名・教室名
What	6	サービスカテゴリ (SDM や空調，照明など)
	7	種別 (立体音響や位置測位など)
	8	型式・機器メーカ
	9	トピック内で一意の識別名称
	10	パラメータ名 (音源操作，温度，湿度など)
入出力	11	(R または W)

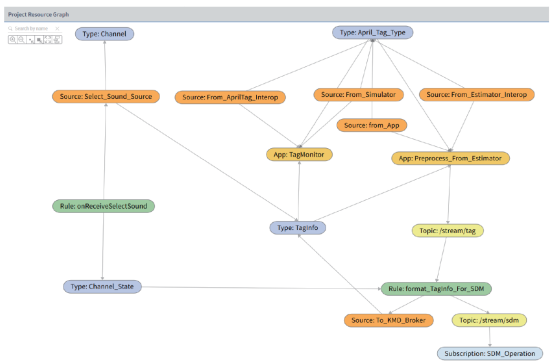

図6　VANTIQによるロジックの記述

カルマンフィルタによる位置推定機能は，Pythonによって実装され，MQTTを介してVANTIQとメッセージ交換している．Pythonとなった理由としては，VANTIQやNodeREDのようなビジュアルプログラミングツールでは，行列計算などがサポートされていないためである．実測値である３次元空間上の位置座標(x,y,z)をフィルタリングによって推定した結果を，図7に示す．カルマンフィルタによって，滑らかに位置座標が推移しているのが分かる．

3.2.2　立体音響の再生

データ変換後のメッセージを解釈・再生するのは，図3右下に示すバンダイナムコスタジオの開発するVSSSである．VSSSはあらかじめ登録された複数の音源を，アプリケーションから指定された三次元位置に提示することができる．VSSSで計算された各スピーカの音声情報は，Dante™規格でヤマハXMV8280-Dへ送信される．XMV8280-Dでは，各スピーカに接続されるアナログケー

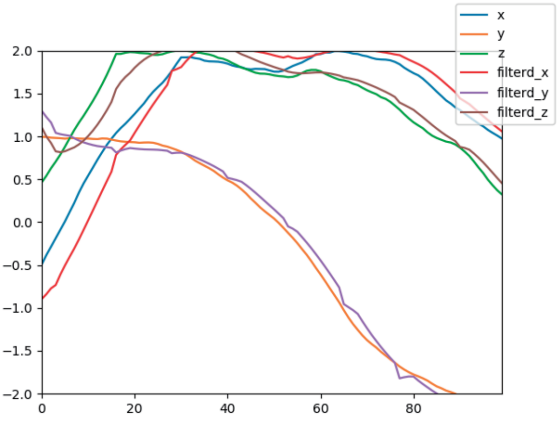

図7　予備実験カルマンフィルタ

*4　https://nodered.org/

ブルを通じて音声情報が伝えられる．こうして，4m四方の角に設置された8つのスピーカ（高さ1.2mと2.7m）を用いてユーザに立体音響を届けることができる．

3.3　リモコン操作・音声可視化アプリケーション

3.1 節に示したインタラクティブ操作アプリケーションとは別に，リモコン操作で音声オブジェクトの位置変更をするためのiPadアプリケーションとして開発した（図8）．一般的なドローン操作に用いられるようなユーザインターフェースを提供しており，画面左側に上昇・下降・右に移動・左に移動の4つのボタンを配置し，画面右側には前進・後退・右旋回・左旋回のボタンを配置した．ユーザの操作に応じて対応するMQTTメッセージが発行されるとともに，プラットフォーム側で処理された位置情報をもとに，ARオブジェクト（画面上ではUFO）が表示さる．アプリによる仮想空間と，立体音響の再生される実空間の座標のキャリブレーションは，感覚的な要素も考慮して，環境ごとに手動で設定することとした．なお，開発環境はUnityである．

第４章　Web360^{2}の開発

インターネット上の動画配信サービスが拡大し続けており，全世界のダウンストリームトラフィックシェアのうち動画ストリーミングのトラフィックシェアが過半数を占めている．VRサービスの普及や民生用の360°カメラの登場などにより3Dコンテンツの視聴体験への敷居が下がり，3Dコンテンツサービスが拡がり始めている．近年，3Dコンテンツサービスをwebブラウザ上で体験するための技術が進み，専用のアプリケーションをインストールする必要もなく，Web上で手軽にサービスを楽しむことができるようになってきた．本研究では，Web上で360°動画およびオブジェクトベースオーディオによる立体音響を再生し，視聴者にインタラクティブな3Dコンテンツの視聴体験を提供するWebアプリケーション「Web360^2」を設計，実装し，さらに評価を行った．Web360^{2}は，GithubPages[*5]にて公開している（詳細は[88]参照）．

図8　リモコン操作・音声可視化アプリの画面

4.1　設計

Web360^{2}のシステム設計概要を図9に示す．視聴者の頭部動作や音響オブジェクトへのタッチ操作を入力として受け付け，操作に応じた映像や音響をシステム内でリアルタイムにレンダリングして出力することでインタラクティブ性を実現する．映像・音響のメディアファイルはそれぞれ独立しており，システム内で同期をとりながら再生する．映像ファイルには無音の360°動画，音響ファイルにはAudioSprite[*6]によって各音源ファイルを1つにまとめた音源群が収録されている．AudioSpriteは複数の音声ファイルを，時間間隔をとりながら並べて，1つの音声ファイルにまとめる技術であり，再生の際のリクエスト数を減らすことができるため読み込み時間を短縮することができる．また，すべての音声が1つのファイルにまとまっていることで，音声ごとの読み込み遅延を気にする必要がなくなり，音声間の同期がとりやすいことも利点として挙げることができる．

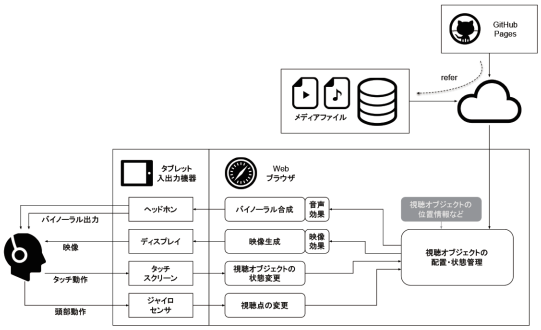

図9　Web360^{2}のシステム設計．

*5　https://sdm-wg.github.io/web360square/
*6　https://github.com/tonistiigi/audiosprite

360°動画はHLSでストリーミング配信され，視聴者を中心とする仮想的な球面内側に投影して再生される．また，音響オブジェクトのもつ3次元位置情報からこの仮想的な球内部の各地点に音響可視化オブジェクトを描画し，各音響の強弱を表現したり，タッチ操作によってON/OFFを切り替えることができる．視聴者の頭部動作は端末に搭載されたジャイロセンサを通してシステムに入力され視聴角度が変更される．視聴角度とON状態にある各音響オブジェクトの位置関係から聴こえるべき音響を生成し出力する．

4.2 素材データ

Web360[2]を開発に用いた素材データは，2016年1月10日に慶應義塾大学日吉キャンパス内に建つ藤原洋記念ホールで開催された慶應義塾大学コレギウム・ムジクム古楽アカデミーのコンサートの収録データ，および，2017年1月26日に六本木ミッドタウン内にあるBillboard Live Tokyoで開催されたMusilogue Bandのコンサートの収録データである．[82]や[79]に，それぞれの収録環境や収録方法が記述されている．

4.3 実装

Web360[2]は，主にWebVR用のフレームワークであるA-FrameとWeb Audio APIを用いて実装した．実装概要を図10に示し，視聴体験中のスクリーンショットを図11に示す．

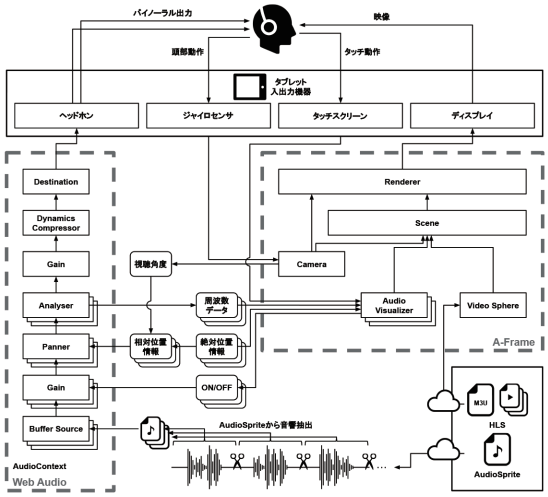

図10 Web360[2]の実装概要．

4.3.1 A-Frameを用いた処理

A-Frameを用いて，360°動画の投影や音響可視化オブジェクトの描画を行う．360°動画はHLSでストリーミング配信され，A-Frameのa-videosphereオブジェクトを用いて，カメラを原点に広がる球面内部に映像を投影する．HLSはSafariやEdge，一部のモバイル端末用ブラウザではネイティブサポートされているが，ChromeやFirefox，IEなどサポートしていないブラウザも多く存在するため，hls.js[*7]を用いて対応した．音響の可視化はWeb AudioのAnalyzerノードから提供される各音響の周波数領域データを用いて表現している．

また，視聴者のインタラクティブな操作は主にAFrameが担う処理を介して入力される．視聴者の頭部動作は端末に搭載されたジャイロセンサに検知され，A-Frameのカメラオブジェクトに入力されることで視聴角度が変更される．タッチ操作はA-Frameでレンダリングしたオブジェクトにタッチすることで入力として認識され，各音響のON/OFF状態を変更することができる．

4.3.2 Web Audio APIを用いた処理

Web AudioのAudioContext上で各ノードを鎖状に繋いでいき，視聴角度とON状態の各音響オブジェクトから聴こえるべき音響をリアルタイムレンダリングする．はじ

図11 Web360[2]のスクリーンショット．

*7 https://github.com/video-dev/hls.js/

めに各音源が1つにまとめられたAudioSpriteのファイルデータから各音源を抽出し，Buffer Sourceノードに入力する．次のGainノードでは，OFF状態の音響のゲインを0にすることで音響のON/OFF状態を表現する．映像は視聴角度と各オブジェクトの絶対座標に応じて，A-Frameが自動的に適切な相対位置を計算しながらレンダリングするが，Web Audio上で音響オブジェクトは視聴角度の変動に自動追従できないため，カメラの視聴角度と絶対座標から相対座標を随時計算してPannerノードに受け渡すことで音響の追従性を実現する．例えば，視聴角度に対して音響オブジェクトが右側に存在すれば右側で音が鳴っているような音響を生成することができる．また，各音響の周波数領域データはAnalyzerノードで計算され，A-Frameの音響の可視化処理に用いる．以上の処理までは各音源ごとにされ，最後にマスターゲインとなるGainノードやDynamics Compressorノードでまとめられバイノーラル出力される．

4.3.3　音響の可視化

Web Audio のAnalyserノードでは，時間領域データやFFT（Fast Fourier Transform，高速フーリエ変換）による周波数領域データを取得することができる．Web360[2]の音響可視化では周波数領域データのみを用いる．音響可視化オブジェクトを構成するA-Frameコンポーネントのtickハンドラを用いて毎秒60回から120回程度のオーダーで，周波数データを取得するAnalyserNode.getByteFrequencyData()メソッドを呼び出す．取得した周波数データの処理行程の概要を図12に示す．

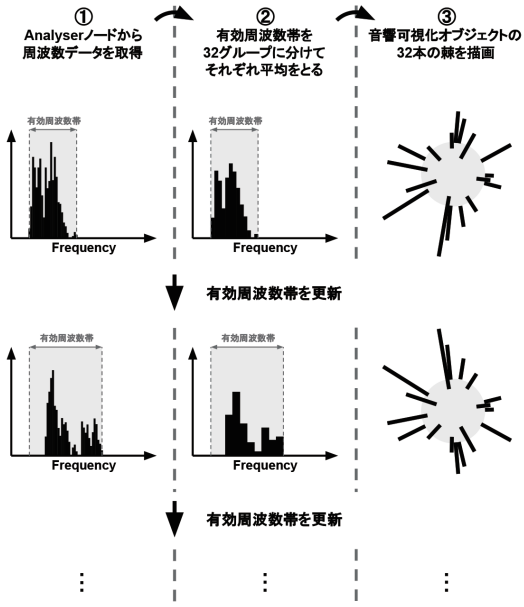

図12　音響を可視化する処理．

AnalyserNode.getByteFrequencyData()メソッドによってナイキスト周波数以下の周波数データが得られるが，極端な高周波数帯・低周波数帯にはあまりデータが存在せず，有効な値が含まれる周波数の範囲は限られていることが多い．そこで，一度でも有効な値が現れたことがある周波数の上限・下限をとり，この範囲を有効周波数帯とする．有効周波数帯は再生開始から周波数データが取得される度に更新される．有効周波数帯の中に含まれるデータを周波数の高さの順に可能な限り均等な32グループに分けて，グループ内で平均値を計算する．得られた32グループそれぞれの平均値は音響可視化オブジェクトの32本の棘の長さと色に変換されて描画される．周波数の高さを音響可視化オブジェクトの棘の長さに対応させて描画してみたところ，近接する周波数グループ同士に相関があるために部分的に棘が盛り上がるような不格好な形状となったため，周波数の32グループと音響可視化オブジェクトの32本の棘の対応関係はランダムに定まるようにした．また，棘の色はOFF状態ではグレーであり，ON状態では平均値が大きくなるにつれて青，緑，黄，オレンジ，赤と変化していく．

4.3.4　映像と音響の同期処理

映像・音響のメディアファイルはそれぞれ独立しているため同期をとりながら再生する必要がある．映像の再生時間はJavaScriptのHTMLMediaElement.currentTimeプロパティによって取得・設定することができる．一方，音響についてはWeb Audio APIに読み取り専用のAudioContext.currentTimeプロパティが用意されているが，これは音響の再生時間ではなくAudioContextが生成されてからの経過時間を示す．音響が停止中であってもAudioContext.currentTimeプロパティの値は単調に増加し続けるため，システム内では音響の再生・停止を管理するWeb Audio APIのAudioBufferSourceNode.start()メソッドおよびAudioBufferSourceNode.stop()メソッドが呼び出される際にAudioContext.currentTimeプロパティの値を記録するなどして音響の再生時間を算出している．

再生開始時やループ再生により先頭に戻った際には映像の再生時間を算出された音響の再生時間に強制的に同期させる．また，音響可視化オブジェクトのAFrameコンポーネントがもつtickハンドラによって毎秒60回から120回程度のオーダーで，算出された音響の再生時間と映像の再生時間の差分を管理する．差分値に対して閾値を設定し，閾値を超えたときに映像の再生時間を音響の再生時間に強制的に同期させる対応は可能ではあるが，同期処理が入るたびに映像再生の滑らかさが失われ，体感品質を損なってしまう．そこで，現行バージョンでは差分の閾値を0.1秒として，閾値よりも映像が遅れたら映像の再生速度を2倍にし，閾値よりも映像が先行したら映像の再生速度を0.5倍にすることで，映像と音響の再生時間のずれが徐々に閾値以内に収束するような比較的制約のゆるい同期をとっている．再生速度の変調は多少の違和感を生じてしまうが，映像再生の滑らかさを保つことができる．映像の再生速度はJavaScriptのHTMLMediaElement.playbackRateプロパティによって設定する．

第５章　co-Soundの開発

インターネットを前提とした視聴サービスは，収録した映像音声データのIPネットワーク化により各システム及びプロセスのソフトウェア制御を行っている．加えて収録対象へのオブジェクトベースによるアプローチは，視聴コンテンツの柔軟な再生を可能としている．一方で三次元映像投影技術の一つである拡張現実（Augmented Reality，AR）は現実空間の要素とデジタル空間情報の両方と対話することを可能としているが，映像音声視聴環境の媒体として活用された事例は数少ない．そこで本研究では，ARを活用した音楽イベントのインタラクティブな映像音声再生アプリケーション「co-Sound」を提案する．co-Soundは，導入コストが低いウェブブラウザ上で，視聴者からの様々な入力に応じて，オブジェクト単位に構築されたARを動的にレンダリングするマルチモーダルなインターフェースとして設計された．さらに複数ユーザ間のARオブジェクト操作をリアルタイムかつ双方向に共有することで，従来1対1に制限されていたユーザとコンテンツとの関係性を拡張し，同一AR空間でのユーザ同士のインタラクションを可能としている．試作したアプリケーションを実装し，AR空間同期の性能評価および被験者からのアンケート評価を行った．複数ユーザからのオブジェクト操作を受け付ける十分な低遅延同期を確認し，WebARを活用したインタラクティブな映像音声視聴メディアとして高い評価を得た（詳細は[89]参照）．

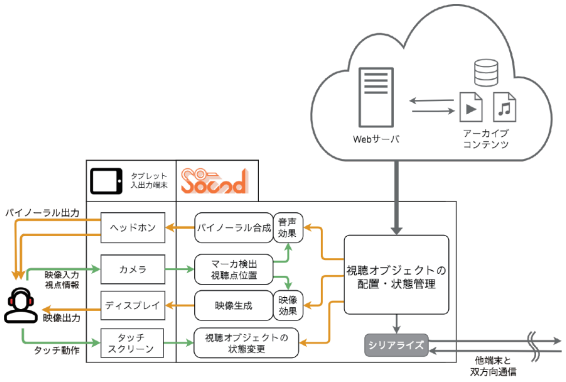

図13　co-Sound設計図

5.1　co-Soundのシステム概要

co-Soundのシステム設計の概要を図13に表す．視聴者はタブレット端末等からARマーカをカメラに写すことにより，co-Soundへ映像情報を入力する．入力された映像情報から，マーカの検出およびカメラ座標の推定が行われ，前節で述べた収録データが持つ位置情報と照合することでARオブジェクトの座標および原点を決定する．またARマーカの検出，AR映像の位置やカメラ座標の推定に加え，視聴者からのタッチ動作を入力情報として，これに応じたAR映像や音響をシステム内でリアルタイムにレンダリングして出力し，インタラクティブ性を実現する．

さらに，co-Sound内で状態管理が行われた視聴オブジェクトのメタデータをシリアライズし，他端末間で相互に通信することで，リアルタイムにAR空間を同期させる．入力となるカメラ映像は端末依存の情報となるため，各ユーザによる自由視点視聴の要求事項は満たされる．

表3　メッセージ同期トポロジの違い

	通信トポロジ	代表的プロトコル	通信量	送信先
P2P	1対1/フルメッシュ	WebRTC	$O(n^2)$	各ピア
client/server	1対1/スター	WebSocket, HTTP	$O(n)$	通信相手
Pub/Sub	1対n/マルチキャスト	MQTT	$O(\log n)$	broker

5.2 リアルタイムなAR空間同期の設計

リアルタイムでAR空間情報を同期するにあたり，表3にていくつかの通信方式を比較する．本研究では低遅延通信方式としてWebRTCを採用する．P2Pによって各ノード間を結びフルメッシュネットワークを構築し，ARオブジェクトのメタデータの同期を行う．同期のための通信量はO(n2)となるが，デバイス間の通信における経由ホップ数が少なく，よりリアルタイム性が保証されることが期待される．

図14 co-Soundスクリーンショット

表4 co-Sound実行環境

WebAR ライブラリ	AR.js v1.5.0, aframe.js v0.9.2
WebGL ライブラリ	Three.js v0.110.0
実行ブラウザ	Chrome v79, Safari v604.1

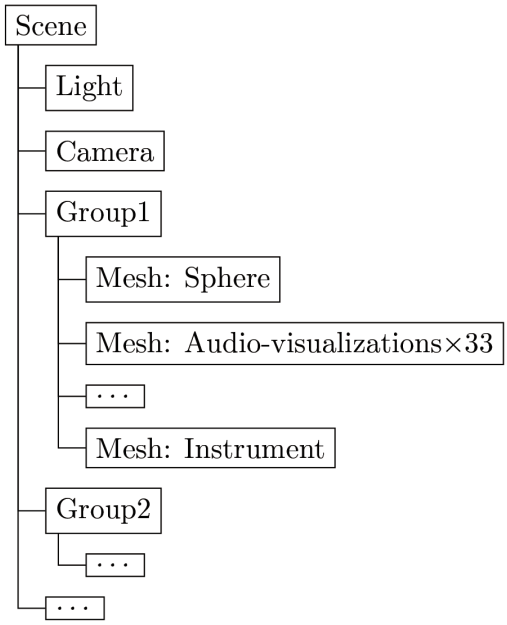

図15 Three.jsオブジェクトのシーン構造

5.3 実装

前節で述べた設計アプローチをもとに，co-Soundを実装した．本システムは，主にWebAR用フレームワークであるAR.js[90]，Webブラウザ上で音声を扱うためのWebAudio API [91]を用いて実装した．アプリケーションのスクリーンショットは図14の通りである

5.4 WebAR動作概要

実行環境は以下の表4の通りである．Webサーバからはco-Soundのアプリケーションデータが送られ，新しくルーム（詳細は5.7節に記述）を設定した場合，【??節】で述べた音楽イベントの基本データである各楽器の3Dモデルファイル，位置情報および映像音声メディアファイルが伝送される．

Webブラウザにおいて，視聴端末（タブレット端末やスマートフォンなど）に付属したカメラからの映像はタグに，生成されるAR映像はタグにレンダリングされるが，マーカの画像認識およびカメラ位置の推定はAR.js，AR映像や音響可視化オブジェクトの描画はThree.jsにより行われる．Three.jsオブジェクトのシーンは図15のように構成し，対象となる楽器ごとに描画グループを分割した．このうちcameraオブジェクトを用いてAR.jsのクラスを定義することで，視聴端末の物理カメラとThree.jsシーン内でのカメラ位置が同期され，任意の位置でAR映像を視聴することが可能となる．

5.5 ユーザインターフェース

co-SoundはARコンテンツの視聴に加え，タブレット端末等を想定し視聴者からのタッチ動作によりARオブジェクトに対し簡易的なインタラクティブ操作を可能とする実装を行った．視聴者はpickingと呼ばれる処理により，視聴端末上のAR映像をタップすることでco-Sound内部オ

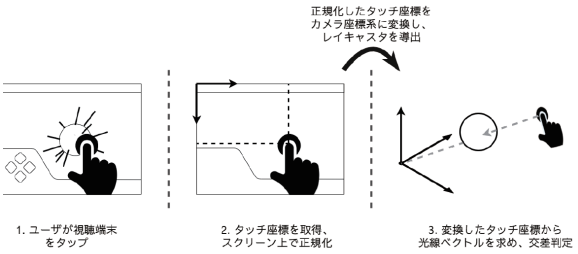

図16 co-Soundにおけるピッキング処理

ブジェクトのデータにアクセス可能となる．co-Soundのpicking処理の概要を図16に示す．

視聴者のタップした座標を，プロパティにより取得し，スクリーン上で-1 – 1へ正規化する．次にこのスクリーン座標系をカメラ座標系に変換する．クラスのメソッドがこれにあたるが，Three.js v0.110.0において正常な計算が行われないため，原義に基づき以下のプログラム1でこれを求めた．更にクラスに，光線が発する元となるカメラの位置ベクトルおよび上式で求めた光線の方向ベクトルを指定する．これによりレイキャスタを作成し，タップした点の位置ベクトルから延長した光線ベクトルを導出する．この光線ベクトルに対しメソッドを利用することで，シーンに存在するオブジェクトとの交差判定を行い，光線ベクトル上のオブジェクトが返される．以上の手順でタップ動作によるARオブジェクトの選択を可能とした．

Listing 1: スクリーン座標系からカメラ座標系への変換

```
const THREE = require('three')

const mat = new THREE.Matrix4();
const mouseVec = new THREE.Vector3(screenX, screenY, 1)
  .applyMatrix4(mat.getInverse(camera.projectionMatrix))
  .applyMatrix4(camera.matrixWorld);
```

選択したオブジェクトに対して，(1)音量のON/OFF; (2)座標の変更; のインタラクションを実装した．目的のARオブジェクトへのタッチ動作により音量のON/OFFが切り替わり，加えて各楽器オブジェクトの中心に位置するSphereオブジェクトが不透明状態となることで，選択状態を表すフラグが与えられたことを示す．フラグが与えられたオブジェクトを操作する十字コントローラ型UI(図14 アプリケーション画面左)を実装し，対象となったAR映像のxyz軸三方向への移動を開発した．

5.6 co-Soundの音響制御

Webブラウザによる音響効果をWebAudioを利用して実装した．AudioContextオブジェクトをベースとして，WebサーバからHTTPリクエストにより取得したBufferSourceから出力であるdistinationまで，ノードを鎖状に連結し音響をリアルタイムレンダリングする．

音響のON/OFF動作は，ゲイン調整ノードであるGainオブジェクトのgain値を0または定数にすることで表現した．Boxオブジェクトを用いた音響の可視化は，Web360^2 [88]同様，WebAudioのメソッドにより時間領域データから周波数領域データを取得し，有効周波数帯をBoxオブジェクトの長さおよび色に変換することで表現した．

最後に，立体音響のリアルタイムレンダリングを実装した．AR映像はAR.jsを用いたマーカー検出およびオブジェクトの位置情報により適当な座標に描画されるが，音像はこれに追従しないため随時座標を計算する必要がある．座標計算に必要な座標系を表5にまとめる．以下では，ある点Pを座標系Aで見たとき，APsubscriptと表記する．AR.jsでは検出したマーカーを原点としたローカル座標系上にAR映像が配置されるため，これをco-Soundのワールド座標系として扱う．すなわちマーカ中心はWO，ある楽器オブジェクトaはWaおよびカメラオブジェクトはWPVと表される．一方，WebAudioにおいて三次元位置情報を扱うPannerノードは視聴者を原点としたローカル座標系を渡す必要があるため，ワールド座標系をビュー座標系に変換する．ビュー座標系から見た楽器aはVq，ワールド座標系の原点はVPWと表され，座標系の回転をVRWと表記すると，以下の式1が成り立つ．

$$\begin{aligned} Vq &= VPW + VRW \cdot Wa \\ &= VRW \cdot (Wa - WPV) \quad (1) \end{aligned}$$

上式よりワールド座標系の任意のオブジェクトをビュー座標系で表せるため，カメラ中心即ち視聴者を基準とした立体音像をレンダリングすることが可能となる．

表5 立体音響のレンダリングに関する座標系

座標系 (および略称)	説明
ローカル座標系 (以下 L)	オブジェクトを原点に取った座標系．
ワールド座標系 (以下 W)	全オブジェクト共通の座標系． AR.js では検出したマーカを基準点とするため，これを原点とした．
ビュー座標系 (以下 V)	カメラを原点に取った座標系．カメラとオブジェクトの相対位置を表す．

5.7 端末間シグナリングおよびコネクション

複数端末間でのAR空間同期のために，リアルタイム対話型マルチメディアサービスとして設計されたオープンソースのPaaS（Platform as a service）であるSkyWay v2.0.1[*8]を利用した．SkyWayはWebRTCのコネクション接続用のシグナリングサーバ，パケットリレー用のTURNサーバ，そしてWebSocketサーバを提供しており，各ピア上で立てたPeerインスタンスから呼び出すことで接続から離脱までのコネクション管理が可能となる．なお，これらのサーバは東京に配備していると公開されている[*9]．本研究ではSkyWayを利用して名前空間で分割されたルームを作成し，co-Soundを立ち上げたピアがルーム上で相互にコネクションを確立するよう設計した．最も有用なケーススタディであるブロードキャスト（すなわち1対nの通信）を用いて，オブジェクトベースに構成されたAR映像音声のメタデータを送信することでリアルアイムな同期を実装した．

ここで，通信プロトコルとして(1)WebRTCによるメッシュ型通信;(2)WebSocketによるスター型通信;の2種類を用いて実装を行った．公開されているSky-Way JavaScript SDK（Software Development Kit）のソースコードでは，ルーム型バイナリデータ通信用にはWebSocketのみ実装されている．そこで1対1型に実装されているWebRTC DataChannelのソースコードをもとに，ルーム型においても各ピア間で相互にDataChannelコネクションを構築するよう改良を行った．

第６章　SDMシンポジウム2019

映像と音声のオブジェクト化が融合することで，従来の配布型コンテンツビジネスを超えた，新しい次元のインタラクティブなオリンピック・パラリンピックの視聴形態などこれまでにないビジネス領域や，これまでにないディジタルメディアを用いた表現方法などを開拓・開花させることが期待されている.2019年7月8日（月）に開催するSDMシンポジウム2019では，以下の関連トピックスに関して，意見交換を行った．

- オブジェクト視聴メディアとソフトウェア制御
- オリンピックに向けた研究開発
- インタラクティブメディアとゲーム
- 新しいメディアデバイス
- シネマ・劇場におけるメディア設備とコンテンツ
- 視聴メディアに関する標準化活動
- 3Dユーザインターフェイス
- 放送に関する研究開発
- 機械学習を利用した時空間情報の解釈と視聴者行動の検知
- 大容量メディアの分散処理とEdge Heavy Computing
- 大容量メディアのためのネットワークとContent Delivery Network

また，パネル討論のパネリストとして以下の6名が登壇した．

- 猿渡洋（東京大学情報理工学系研究科教授）
- 池田雅弘（ヤマハ株式会社音響事業本部開発統括部長）
- 光藤祐基（ソニー（株）RDセンター基盤技術研究開発第1部門オーディオ技術開発部2課統括課長）
- 村井幹司（株式会社ジェネレックジャパン 代表取締役）
- 山下修（株式会社オーディオブレインズ 技術部 アプリケーションサポート課）
- 塚田学（東京大学情報理工学系研究科准教授）

また，SDMシンポジウム2019の詳細として，プログラムを含む，ウェブサイト情報を付録Aに添付した．

第７章　まとめ

本報告書では，2014年より開始したSDMコンソーシアムで進めて来た，視聴空間サービスのソフトウェア制御による研究で，2019年度に行なった活動について報告した．

---

*8 https://github.com/skyway/skyway-js-sdk
*9 https://support.skyway.io/hc/ja/articles/115003112067-TURN サーバの配置場所について（Accessed on 01/04/2020）

SDMコンソーシアムでは，ソフトウェア処理による視聴空間の制御，映像音声を制御するネットワーク機器，インタラクティブ・ユーザインターフェイス，SDMプラットホームを利用したコンテンツ作成など，共同研究活動に参加するパートナーを募集しています．

ご協力をよろしくお願いいたします．

Appendix　SDM シンポジウム2019の詳細

SDM

戻る

Software Defined Media シンポジウム2019

2019.07.08（月）13:00-17:00
東京大学 I-REF棟 6F ヒロビー

SDMシンポジウム2019

映像と音声のオブジェクト化が融合することで，従来の配布型コンテンツビジネスを超えた，新しい次元のインタラクティブなオリンピック・パラリンピックの視聴形態などこれまでにないビジネス領域や，これまでにないディジタルメディアを用いた表現方法などを開拓・開花させることが期待されている.2019年7月8日（月）に開催するSDMシンポジウム2019では，以下の関連トピックスに関して，意見交換を行う.

- オブジェクト視聴メディアとソフトウェア制御
- オリンピックに向けた研究開発
- インタラクティブメディアとゲーム
- 新しいメディアデバイス
- シネマ・劇場におけるメディア設備とコンテンツ
- 視聴メディアに関する標準化活動
- 3Dユーザインターフェイス
- 放送に関する研究開発
- 機械学習を利用した時空間情報の解釈と視聴者行動の検知
- 大容量メディアの分散処理とEdge Heavy Computing
- 大容量メディアのためのネットワークとContent Delivery Network

プログラム

13:00 - 13:10	開会挨拶		江崎浩
13:10 - 13:40	教師無し学習に基づく自律的な音メディア信号処理とその応用		猿渡洋
13:30 - 13:50	Software Defined Media：視聴空間サービスのソフトウェア制御		塚田学
13:50 - 14:20	Software Defined Sound Field：「SURROUND:AI」と「AFC – Active Field Control」		池田雅弘
14:20 - 14:50	Sonic Surf VR: 音のVRを実現する波面合成技術とコンテンツクリエーションについて		光藤祐基
14:50 - 15:10	高品位なイマーシブオーディオ再生環境について（スピーカーアラインメント技術）		村井幹司
14:50 - 15:10	ASTRO SPATIAL AUDIO：ライブエンターテイメントで実用化されるオブジェクトベースイマーシブソリューション		山下修
	休憩・デモンストレーション展示の見学　（15:30 - 15:40）		
15:40 - 17:00 (80分)	パネル討論（モデレータ：江崎・砂原）		
17:00	閉会挨拶		

講演者情報

江崎 浩
東京大学 情報理工学系研究科 教授

昭和 38 年生.昭和 62 年九州大学・工・ 電子修士課程了.同年 (株) 東芝入社.平 成 2 年米国ニュージャージ州ベルコア社. 平成 6 年コロンビア大学・客員研究員.平 成 10 年東京大学大型計算機センター・助 教授.平成 13 年同大学大学院・情報理工学系研究科・助教授.平成 17 年同大学大学院・同研究科・教 授,現在に至る.博士 (工学,東京大学).MPLS-JAPAN 代 表,IPv6 普及・高度化推進協議会専務理事,WIDE プロジェ クト代表,JPNIC 副理事長.

猿渡 洋
東京大学 情報理工学系研究科 教授

Hiroshi Saruwatari received the B.E., M.E., and Ph.D. degrees from Nagoya University, Japan, in 1991, 1993, and 2000, respectively. He joined SECOM IS Laboratory, Japan, in 1993, and Nara Institute of Science and Technology, Japan, in 2000. From 2014, he is currently a Professor of The University of Tokyo, Japan. His research interests include audio and speech signal processing, blind source separation, etc. He received paper awards from IEICE in 2001 and 2006, from TAF in 2004, 2009, 2012, and 2018, from IEEE-IROS2005 in 2006, and from APSIPA in 2013 and 2018. He received DOCOMO Mobile Science Award in 2011, Ichimura Award in 2013, The Commendation for Science and Technology by the Minister of Education in 2015, Achievement Award from IEICE in 2017, and Hoko-Award in 2018. He has been professionally involved in various volunteer works for IEEE, EURASIP, IEICE, and ASJ. He is an APSIPA Distinguished Lecturer from 2018.

池田雅弘
ヤマハ株式会社 音響事業本部開発統括部長

1993年京都大学工学部建築学科卒，95年同大学大学院博士前期課程工学研究科建築学専攻修了，同年ヤマハ株式会社入社．以降，同社の研究開発部門・事業部門にて建築音響・空間音響・デジタル商品プラットフォーム・プロオーディオ関連の研究開発・事業展開に従事．ヤマハミュージックアジア（シンガポール）社長を経て現職．
日本音響学会，ＡＥＳ，米国音響学会，各会員

光藤 祐基
ソニー（株）R&Dセンター基盤技術研究開発第1部門
オーディオ技術開発部 2課 統括課長

2004年入社。Walkman、VAIO向けの高音質化技術の開発に携わる。2011年にフランスの国立音響音楽研究所IRCAMにて客員研究員として滞在し、音楽を楽器ごとに分離する技術「音源分離」の研究開発を行う。帰国後は音のVRとも呼ばれる波面合成技術のリーダーとして、B2B事業部とともにSonic Surf VRを立ち上げる。現在、IEEE ICCE国内プログラム委員、音源分離国際コンペティションSiSECのGeneral Chairを務める。

山下 修
株式会社オーディオブレインズ　技術部　アプリケーションサポート課

2000年法政大学 法学部卒業。ヒビノ株式会社プロオーディオセールスDiv入社。2009年株式会社オーディオブレインズ入社。現在はアプリケーションサポート課としてシステム設計、現場の調整、現場サポートなどを主に活動。公益社団法人日本舞台音響家協会、公益社団法人劇場演出空間技術協会音響部会の構成メンバーとしても活動中。2018年よりオーディオブレインズ輸入品としてオランダASTRO SPATIAL AUDIOの輸入事業に着手。ライブエンターテイメント環境でのイマーシブソリューションの実現に取り組む。

村井幹司
株式会社ジェネレックジャパン　代表取締役

2004年4月株式会社エムアイセブンジャパン入社ジェネラルマネージャーを経て取締役代表執行役、2017年7月よりモニタースピーカーの世界的メーカーであるGENELEC OYと国際合弁の株式会社ジェネレックジャパン設立、イマーシブオーディオの音楽体験の機会や製作から再生に関するセミナーやワークショップを2014年から行い情報共有をすすめている。

塚田学
東京大学 情報理工学系研究科 准教授

2005年慶應義塾大学環境情報学部卒業。2007年慶應義塾大学政策・メディア研究科修士取得。2007年よりフランス・パリ国立高等鉱業学校 (Mines ParisTech) ロボット工学センター博士課程在籍および、フランス国立情報学自動制御研究所 (Inria)にて研究員として勤務。2011年博士号取得。 現在は、東京大学 大学院 情報理工学系研究科の准教授。2014年よりWIDEプロジェクトのボードメンバー、およびSoftware Defined Media（SDM）コンソーシアム・チェア。 自動車の情報化など、次世代インターネットIPv6における移動体通信、SDMに取り組む。

過去のSDMシンポジウム

SDMシンポジウム2017

2017年7月20日に東京大学 I-REF棟6Fにて開催されたSDMシンポジウム2017では、産学からゲストスピーカを招き活発な意見交換をしました。また、外部のパネリスト9名を交えたパネル討論を行いました。

2017年7月20日

Webページ

SDMシンポジウム2016

2016年7月29日に東京大学 I-REF棟6Fにて開催されたSDMシンポジウム2016では、産学から5名のゲストスピーカを招き活発な意見交換をしました。また最新技術のデモンストレーション展示も行われました。

2016年7月29日

Webページ

A　SDM シンポジウム 2019 の詳細

受付は終了しました。

シンポジウム参加受付
(2019.07.08)

Software Defined Mediaコンソーシアムは、2014年に開始されたインターネットメディアを研究し、ビジネス機会の創造を行う産学コンソーシアムです。

Webページ

アクセス

住所： 〒113-8657 東京都文京区弥生1-1-1 東京大学 I-REF 棟6F
メール： sdm-wg-request@wide.ad.jp

地図