co-Sound: Web AR を用いたインタラクティブメディア基盤の設計と実装

井口 和真[†1]　塚田 学[†1]　江崎 浩[†1]

概要：インターネットを前提とした視聴サービスプラットフォームは，収録した映像音声データの IP ネットワーク化によりネットワーク上に分散して存在するシステム及びプロセスを，ソフトウェアにより管理・制御する．さらに収録対象のオブジェクトベースによるシステムの設計と実装によって，視聴コンテンツの柔軟な再生を可能とする．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内の要素の両方と対話することを可能としているが，映像音声視聴プラットフォーム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 空間での複数のユーザ間で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が実現されていることを確認し，さらに，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ることができた．

co-Sound: The interactive media platform with Web AR

KAZUMA INOKUCHI[†1]　MANABU TSUKADA[†1]　HIROSHI ESAKI[†1]

1. はじめに

近年インターネットを前提とした映像音声視聴サービスプラットフォームが注目を集めている．現在主流の映像コンテンツは静的な視聴体験にとどまっている一方，収録から再生のプロセスを仮想化・抽象化した映像音声視聴システムは，再生環境の自由な設計が可能となる．収録した映像音声データの IP ネットワーク化により，ネットワーク上に分散して存在するシステム及びプロセスをソフトウェアで管理・制御し，オブジェクトベース音響システムを基盤として，複数のオブジェクトとして分割し収録した対象を，再生側でメタデータをもとに空間を再構築する．ビットマップ映像情報と 2 チャンネル音声情報として平面的に捉えるチャネルベースや，HOA (High-order Ambisonics，高次アンビソニックス) と呼ばれる球面調和関数の数学的解析による音場再現を用いたシーンベースとは異なり，映像表現に三次元空間技術を用いることで，現実空間に存在する視聴対象を解釈・表現可能となる．

三次元映像表現の手法としてディジタル環境と実世界の環境を組み合わせた技術は，総称して XR (cross Reality) と呼ばれる．XR 市場は年率 78%で成長し，2023 年には 17 兆円規模に達すると予測されており [1]，アカデミックの分野でも日本バーチャルリアリティ学会[*1]をはじめ研究が進められている．

AR (Augmented Reality) は現実の風景にディジタルコンテンツを重ね合わせる技術であり，ユーザの視界を支配する VR とは異なり物理世界とディジタル表現されたオブ

[†1] 現在，東京大学
Presently with The University of Tokyo

[*1] 入手先 ⟨https://vrsj.org/⟩ (Accessed on 01/05/2020)

ジェクトを合成提示することが可能である．さらに，コンピュータビジョン・画像処理分野の急速な発展により機械学習を利用したリアルタイムな画像認識が可能となった．これにより AR 視聴体験のリアルタイム性やパフォーマンスが改善されている．従来のマーカー型 AR で用いられていた単純な平面画像認識だけでなく，立体認識や Visual SLAM の技術により，マーカーレス型と呼ばれる，画像情報のみによる AR オブジェクトをリアルタイムで高精度な特徴点＋特徴点周辺の 3 次元構築という形で表現可能となっている．エンターテインメント性の高いコンシューマ向けコンテンツだけでなく，医療や教育分野での活用 [2] も提案されている．

2. 本研究の目的

本研究では，AR，特に WebAR を用いた音楽イベントの再生メディア方式を検討する．音楽イベントの収録–再生の一連のプロセスを対象として，ユーザとシステム，ユーザとユーザがインタラクティブに体験可能な視聴，すなわち視聴者から入力された動作を受け付けたインタラクティブな再生，かつ視聴者間で双方向にコミュニケーション可能な映像音響システムの構築を行う．また，実際の音楽イベントの収録データを利用した WebAR アプリケーションを試作する．これらのシステム設計にあたり，要求事項は以下の通りになる．

視聴者 – コンテンツ間におけるインタラクティブな視聴

収録された音楽イベントを AR 上に展開する．実空間から完全に分割された VR ではなく親和性の高い AR を用いることで，実空間に合わせたディジタル空間の投影が可能となる．また視聴者からの動作に応じた個別の映像音源データへのアクセスを許可し，従来の静的なコンテンツ再生では行えない音像の選択・再配置など視聴者自身の動的な再生および編集メディアを実現する．映像表現に加え，音声表現も立体音響での提示を目指し，視聴者の動作に追従し没入感・臨場感を与える効果が求められる．

視聴者 – 視聴者間における双方向コミュニケーション

従来方式は視聴者とコンテンツが 1 対 1 に限定されたものであったが，AR が描画されるディジタル空間を複数人で共有・同期することで新たなインタラクティブ性を創造する．メディアデータをインターネット上で管理し，ソフトウェア制御を行う視聴メディアにおいて，視聴者は受信者としてだけでなく，コンテンツの送信者としての役割も担える．メディアは単純な再生機器ではなく複数人からの操作を受け付けリアルタイムに提示可能な機能が求められる．AR メディアデータのオブジェクトベース構造化 AR コンテンツ群をオブジェクトベースな映像音声データに構造化し，SDM アーキテクチャに基づいたソフトウェア制御処理を可能とする．上述の立体音響も，チャンネルベースや HOA/アンビソニックスではなくオブジェクトベースで設計することにより容易な個別の音源制御も期待できる．

視聴デバイスに依らない柔軟な AR 視聴体験

提案システムは Web アプリケーションで実装する．Web アプリケーションは専用のアプリケーションのインストールを必要とせず，Web ブラウザから直接利用可能な手軽さのため視聴者へのコンテンツ導入コストが非常に低い．ブラウザ組み込み型 WebXR はアプリケーション型やハードウェア型と同様にクロスプラットフォームの点で劣っている．OS や端末に依存しており，視聴者だけでなく再生環境の開発者にとってもコストが大きいため，Pure な Frontend で設計した WebAR アプリケーションを採用するのが適当である．

3. 関連研究

3.1 SDM 試作システム

映像音声コンテンツ，特に現実に存在する対象を三次元的に解釈・表現可能としたインタラクティブな視聴プラットフォームを構築する取り組みとして SDM コンソーシアム [3] が活動している．

SDM は図 1 に示される SDM アーキテクチャを構成し，以下に挙げられる要求事項を満たす．

(1) 三次元の映像音声演出のソフトウェア制御
(2) ソフトウェアレンダリングによる拡張演出
(3) 複数収録対象のミキシング可能性
(4) ユーザ・インタラクション性

LiVRation [4] は，収録された音楽イベントをヘッドマウントディスプレイを用いて自由視点からインタラクティブ再生することを目的としたシステムである．複数地点から 360 度カメラで映像を，指向性/無指向性マイクで音声を収録した実際の音楽イベントを，オブジェクトベースオーディオの手法をもとに三次元映像音声として仮想空間内にレンダリングした．また Web360^2 [5] は，LiVRation と異なり，WebVR を用いてタブレット端末等のブラウザ上で 3D コンテンツを視聴することを前提としている．いずれも視聴者の頭部動作やタッチ動作に追従してコンテンツをインタラクティブに提示するものであり，7 段階のリッカート尺度による被験者実験においても，最高評価の 7 および次点の 6 を合わせた回答数は全回答数の半数以上を占めている．

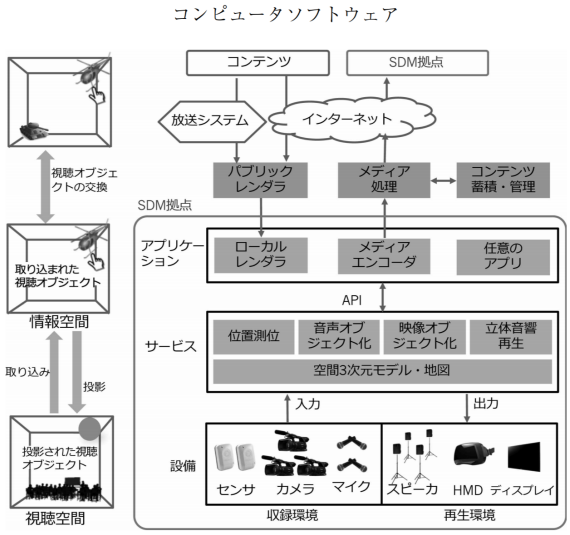

図 1 SDM アーキテクチャ (Fig.1 [3])

3.2 視聴メディアとしての AR

近年博物館や美術館において，AR を展示品の視聴メディアとして利用するユースケースが増加している．Fenu らの研究 [6] では，Svevo 博物館に訪れた 34 人の被験者を対象に，AR を用いてスマートフォンで博物館内を鑑賞してもらった上で，被験者の行動記録と体験後の 5 段階評価付けを分析した．全体的な満足感，経験の新規性，美的感覚，内容への関心等のパラメータで非常に高いスコアを記録し，AR による展示物の鑑賞は有用であることが示されている．また Tillion ら [7] は，美術鑑賞の活動を (i) *Analytical Activity*; (ii) *Sensitive Activity*; に分類し，AR ガイドを用いた鑑賞が各々の活動にどのような影響を与えるか調査した．AR ガイドは，美術品への没入感を阻害し Sensitive Activity に負の影響をもたらすこともあるが，適切な情報の提示 (絵画の素材や他の芸術作品の紹介など) は訪問者にとって Analytical Activity を促進させる可能性を持つ，といった結論を得ている．

以上のような AR を媒体とした美術品・芸術品鑑賞の研究は一定の有用性を提示している一方，オーディオビジュアルコンテンツ及びその AR 視聴に注目した研究はほとんど行われていない．

4. 提案と設計

本研究では，第 2 節で挙げた要件を満たすものとして，co-Sound アプリケーションを提案する．co-Sound は，収録された音楽イベントを対象とし，立体音響効果および複数端末間での空間同期が可能なマーカ型 WebAR アプリケーションである．視聴者はブラウザで起動する WebAR を介しマーカ上に存在する音楽シーンを模した 3D オブジェクトを自由視点で視聴可能，かつその視聴位置での立体音響空間を体験できる．また特定のルーム内に参加したピア同士で AR オブジェクト情報を双方向で通信し合うことにより，複数端末間でのリアルタイムな空間同期を可能としている．

4.1 収録データ

co-Sound アプリケーションで利用した音楽イベントは，2017 年 1 月 26 日に六本木ミッドタウン内 Billboard Live Tokyo で開催された Musilogue Band のコンサートを収録したものである．このコンサートでは，Drums，Electric Bass，Keyboard の 3 種からバンドが構成されており，各楽器単体でマイクを設置し，個別に音源を収録した [8]．なお本研究では，SDM Ontology[9] で定義される音楽イベントのデータ構造のうち，Target 情報を対象とし，上述のコンサートにおける各楽器の属性，位置情報および音声情報を利用した．

4.2 システム設計

co-Sound のシステム設計の概要を図 2 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で，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を行い，前節で述べた収録データが持つ位置情報と照会することで AR オブジェクトの原点および各座標を決定する．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が満たされる．

リアルタイム同期のための AR 空間情報の通信方式は WebRTC を採用する．P2P によって各ノード間を結んだ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 $O(n^2)$ 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，より低遅延性が期待される．

4.3 システム実装

実装概要

前節で述べた設計アプローチをもとに，co-Sound を実装した．実装したアプリケーションの画面は図 3，実行環境は表 1 の通りである．Web サーバより co-Sound アプリケーションがブラウザ上で構築され，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，マーカの画像認識およびカメラ位置の推定

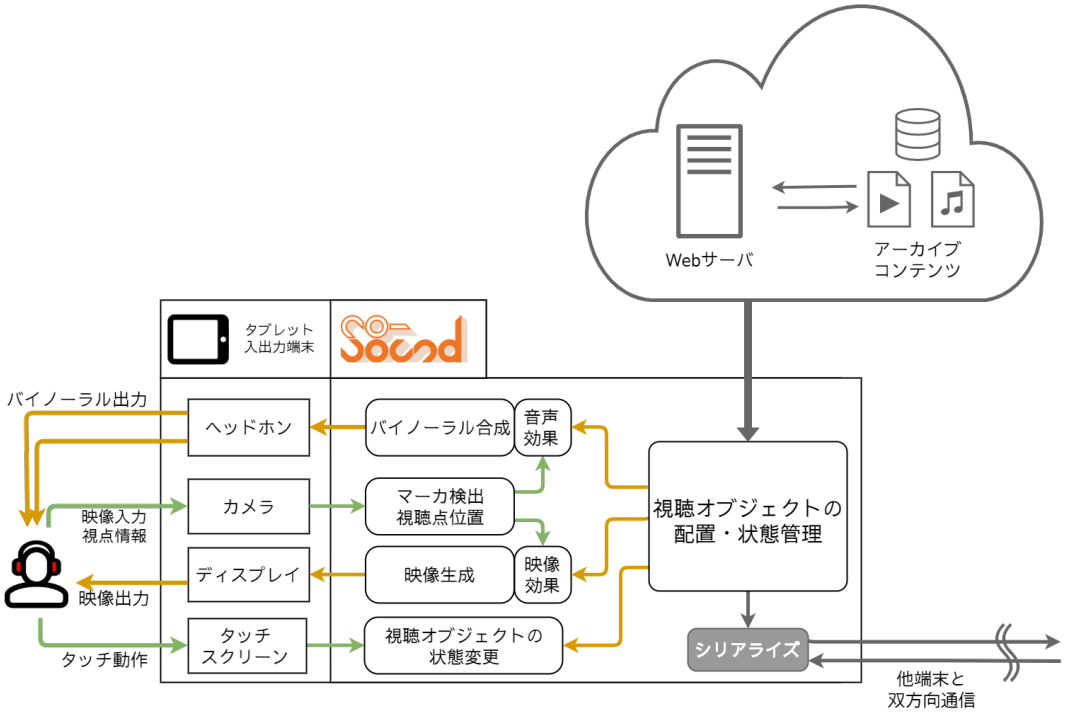

図 2　co-Sound 設計概要図



図 3　co-Sound スクリーンショット

は AR.js[*2] および aframe.js[*3]，AR 映像や音響可視化オブジェクトの描画は Three.js[*4] により行われる．対象となる楽器ごとに描画グループを分割し，camera オブジェクトを用いて AR.js の ArMarkerControls 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．

インタラクション

co-Sound は 視聴端末としてタブレット等を想定し，視聴者からのタッチ動作による AR オブジェクトへの簡易的なインタラクティブ操作を可能とした．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 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．

表 1　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

*2 入手先 ⟨https://github.com/jeromeetienne/AR.js⟩(Accessed on 01/05/2020)
*3 入手先 ⟨https://aframe.io/blog/arjs/⟩(Accessed on 01/05/2020)
*4 入手先 ⟨https://threejs.org/⟩(Accessed on 01/05/2020)

なお視聴者は 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 内部オブジェクトのデータにアクセス可能となる．

WebAudio を用いた音響

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 Buffer Source 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] [5] 同様，WebAudio の`AnalyzerNode.getByteFrequencyData()`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．

カメラ座標を基準とした立体音響

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にまとめる．以下では，ある点 P を座標系 A で見たとき，${}^{\mathrm{A}}\boldsymbol{P}_{\mathrm{subscript}}$ 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 ${}^{\mathrm{W}}\boldsymbol{O}$，ある楽器オブジェクト a は ${}^{\mathrm{W}}\boldsymbol{a}$ およびカメラオブジェクトは ${}^{\mathrm{W}}\boldsymbol{P}_V$ 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 ${}^{\mathrm{V}}\boldsymbol{q}$，ワールド座標系の原点は ${}^{\mathrm{V}}\boldsymbol{P}_{\mathrm{W}}$ と表され，座標系の回転を ${}^{\mathrm{V}}\boldsymbol{R}_{\mathrm{W}}$ と表記すると，以下の式 1 が成り立つ．

$$\begin{aligned} {}^{\mathrm{V}}\boldsymbol{q} &= {}^{\mathrm{V}}\boldsymbol{P}_{\mathrm{W}} + {}^{\mathrm{V}}\boldsymbol{R}_{\mathrm{W}} \cdot {}^{\mathrm{W}}\boldsymbol{a} \\ &= {}^{\mathrm{V}}\boldsymbol{R}_{\mathrm{W}} \cdot ({}^{\mathrm{W}}\boldsymbol{a} - {}^{\mathrm{W}}\boldsymbol{P}_{\mathrm{V}}) \end{aligned} \tag{1}$$

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．

端末間コネクション

複数端末間でのコネクションのために，WebRTC [10] を利用した AR 空間情報の同期を行う．WebRTC とは，ウェブブラウザやモバイルアプリケーションにおいてシンプルな API 経由で P2P リアルタイム通信を可能とする技術である．バイナリデータ通信用の data channel は，メッセージ指向で多重ストリームの機能を持つ SCTP (Stream Control Transmission Protocol) を採用しており，メッセージの到着保証および順序性を任意に行うことができる．一般に，信頼性のあるモードにおいてパケットロスが多く発生する場合，これらの処理にオーバーヘッドが生じるため潜在的に動作が遅くなるが， WebRTC は NAT 越えを行うにあたり SCTP over DTLS over UDP の形式を採っており，実質的に SCTP では輻輳制御とフロー制御を，NAT 越えのトランスポート機能は UDP に担っている．すなわち再送処理・順序保障を行う TCP をベースとする，リアルタイムなデータ送信制御プロトコルの RTSP (Real Time Streaming Protocol) を用いる場合よりもパケット伝送速度が速い [11].

実装には，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 [*5]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 [*6]．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．

なお，比較実験を行うために (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 種類の通信方式ならびにプロトコルを用いて実装を行った．公開されている SkyWay 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．

5. 評価

5.1 性能評価

以降では n 番目のピアを P_n と表す．

以下の実験では，RTT (Round Trip Time) を計測することでこの AR 空間同期の遅延を評価した．P_1 で生成さ

[*5] 入手先 ⟨https://github.com/skyway/skyway-js-sdk⟩(Accessed on 01/05/2020)

[*6] 入手先 ⟨https://support.skyway.io/hc/ja/articles/115003112067-TURN サーバの配置場所について ⟩(Accessed on 01/05/2020)

表 2　co-Sound 実験環境

	OS / CPU / メモリ
PC	Windows 10 version 1809 / Intel® Core™ i7-8550U / 16 GB
タブレット	iOS 12.3.1 / Apple A10X Fusion / 4 GB
スマートフォン	Android 9, EMUI version 9.1.0 / HiSilicon Kirin 960 / 4 GB

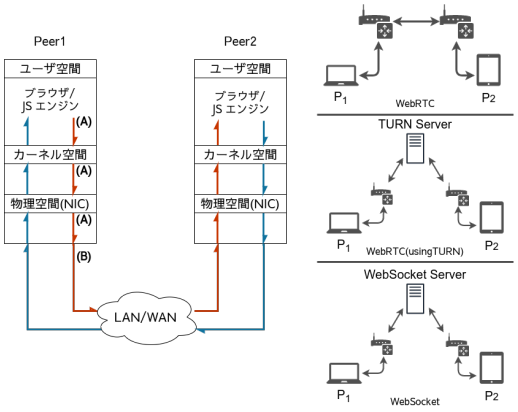

図 4　端末間通信トポロジ

れたパケットを，ブラウザで送信命令を出してから P_1 のユーザ空間，カーネル空間，NIC (Network Interface Card) を経てインターネットに送出されるまでの内部処理時間を t_1^A，LAN (Local Area Network) または WAN (Wide Area Network) を経由して P_1 から P_2 までへの転送にかかる時間を t_{12}^B と表すとする．P_2 は P_1 から受信したパケットをブラウザまで渡し，そのまま同じパケットを P_1 へ送り返す．ブラウザ上でのパケットの受送信にかかる処理時間を t_2^P とする．以上より一連のプロセスによる時間 RTT を表すと，式 2 の通りである．

$$\mathrm{RTT} = 2 \times (t_1^A + t_{12}^B + t_2^A) - t_2^P \tag{2}$$

実験に用いた端末は表 2 の通りである．なお，ブラウザのバージョンは表 1，利用する WebSocket サーバ，Signaling サーバおよび TURN サーバは NTT Enterprise Cloud[*7]が提供するものである．本測定において利用した各サーバのパケット処理性能の違いにより遅延が増減する可能性については考慮しないものとする．WebRTC による通信は信頼性モードを ON にしている．

実験 1–4 の結果を図 5–8 に示す．

実験 1: 通信プロトコルによる遅延

実験 1 では，AR オブジェクト情報のピア間通信に用いるプロトコルをパラメータとして，co-Sound がより低遅延で同期可能であることを示す．比較する通信プロトコルは Web ブラウザで利用可能なリアルタイム通信プロトコルとして (1) WebRTC による LAN 内 P2P 通信; (2) TURN サーバを利用した際の WebRTC 通信; (3) WebSocket 通信; の 3 種類を選択した．また P_1 として表 2 の PC を，P_2 としてタブレットを選択し，1KiB の JSON データを 5 秒間隔で 100 回転送させた際の RTT を計測した．

図 5 より，端末間通信プロトコルとして WebSocket を選択したときの平均 RTT は 210 ms，WebRTC (host) を選択したときの RTT は 73 ms と，65.0%短縮された．WebRTC (relay) のときも平均 RTT は 107 ms となった．これより，提案手法となる WebRTC による端末間通信は，パケットリレーを行った際でも WebSocket 通信と比較して低遅延でパケット転送が可能である．また標準偏差は同順に 116 ms，47 ms，87 ms と導出されたことから，遅延時間のバラつきも抑えられた．

実験 2: メッセージサイズによる遅延

実験 2 では，メッセージサイズをパラメータとした際の転送遅延について評価を行う．メッセージサイズは昇順に 20bytes，120bytes，220bytes，420bytes，820bytes，1KiB，2KiB および 4KiB である．各々のメッセージを転送し合った際の RTT を計測した．実験 1 同様，P_1 として表 2 の PC を，P_2 としてタブレットを選択し，5 秒間隔で 100 回転送させた．

メッセージサイズを 20 bytes から 4096 bytes まで変更させた場合は，図 6 より WebRTC・WebSocket ともに平均 RTT・標準誤差に大きな変化は見られなかった．20–4096 bytes の間では，WebRTC の平均 RTT は約 80 ms，WebSocket の平均 RTT は約 200 ms と，メッセージサイズに依らずおよそ一定であった．

実験 3: 接続端末数による遅延

実験 3 では，接続端末数をパラメータとした際の転送遅延について評価を行う．表 2 の PC，タブレットに加え同表のスマートフォン 1 – 3 台を同一ルーム内に参加させ (これらを $P_3, ..., P_5$ とする)，WebRTC または WebSocket の 2 通りで接続した．5 秒間間隔で 1KiB の JSON データを 100 回転送させ，RTT を計測した．

接続端末数を P_1 と P_2 の 2 台から P_5 までの 5 台に増やしたときの平均 RTT は，図 7 より WebSocket のとき 65 ms，WebRTC のとき 170 ms であった．小規模では接続台数が増加してもピア間の遅延には影響が無いと言える．

実験 4: 端末性能による遅延

実験 4 では，接続端末の性能差による転送遅延への影響について評価を行う．P_1 には表 2 の PC を用い，P_2 とし

[*7] 入手先 ⟨https://ecl.ntt.com/documents/service-descriptions/webrtc/webrtc.html⟩(Accessed on 01/05/2020)

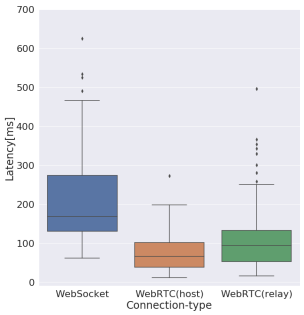

図 5　実験 1: 通信プロトコル別 RTT 測定の結果

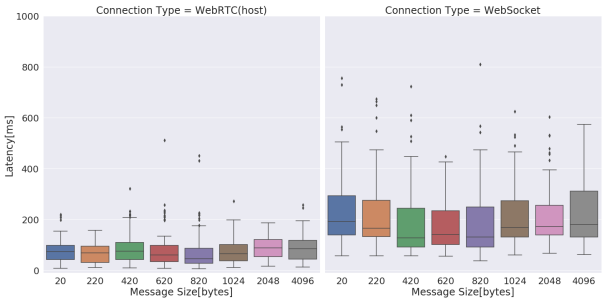

図 6　実験 2: メッセージサイズ別 RTT 測定の結果

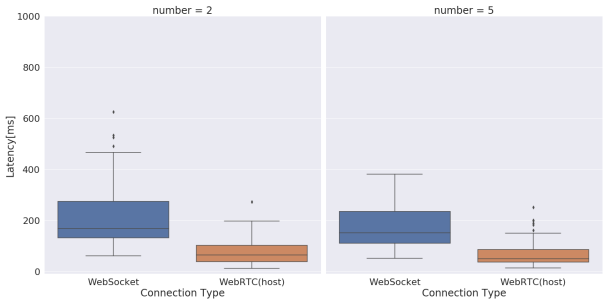

図 7　実験 3: 接続端末数別 RTT 測定の結果

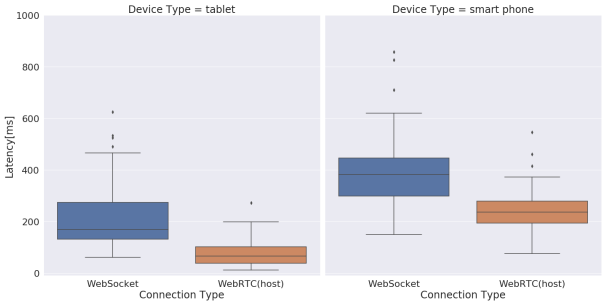

図 8　実験 4: 端末性能別 RTT 測定の結果

てタブレットまたはスマートフォンを選択した際の RTT を計測し，比較した．通信プロトコルには WebSocket または WebRTC を用い，5 秒間隔で 1KiB の JSON データを 100 回転送させた．

実験 4 の測定より，表 2 のスマートフォンを選択した場合，平均 RTT は WebRTC では約 240 ms，WebSocket では約 360 ms となった．図 8 より，ピアとして用いる端末性能は，WebSocket・WebRTC ともに遅延に大きく影響を与えると言える．

5.2　主観評価

定量的評価に加え，アンケート調査により co-Sound の主観評価を行った．

アンケート収集

12 月 6 日から 17 日にかけて co-Sound を体験してもらった被験者にアンケートを回答してもらい，主観評価の調査を行った．視聴端末として Apple iPad Pro (10.5 inch) iOS 12.3.1，ヘッドホンとして Sony WH-1000XM2 を利用し，有線接続して体験を行った．はじめにアプリケーション概要と操作方法を説明し被験者に自由に視聴体験を行ってもらった後に，アンケートの回答を求め評価を取得した．体験者の総数は 25 名，内男性 24 名，女性 1 名であった．年齢構成は 20 代 20 名，30 代 2 名，40 代 1 名，50 代 2 名であり，職業構成は学生 21 名，教員 2 名，社会人 2 名となった．

評価項目

アンケートは，以下に示す 8 項目から成る．それぞれ 1 を最低評価，7 を最高評価とした 7 段階のリッカート尺度を用い，co-Sound 体験後に適当な評価付けを行ってもらった．またアンケートの最後には自由記述欄を設け，体験に関するコメントを得た．

Q1　音声は AR 映像の方角から聴こえましたか？
Q2　音声は AR 映像の距離感と一致していましたか？
Q3　AR 映像を動かしたとき，音響も追従して動いたと感じられましたか？
Q4　AR マーカに対して自分の位置を変えたとき，音響も追従して動いたと感じられましたか？
Q5　音量可視化による音響オブジェクトの ON/OFF 操作は直感的でしたか？
Q6　コントローラによる AR 映像の移動は直感的でしたか？
Q7　AR マーカの認識精度は十分でしたか？
Q8　Web ブラウザ上でも 3D コンテンツのインタラクティブな視聴体験ができましたか？

設問 Q1 – Q4 は音響について問う設問である．加藤らの先行研究 [5] においても音響の立体感に関する評価を行ったが，評価が分散していた．これは音響の立体感を構成する要素のうち，いずれを問うた設問であるかが被験者にとって曖昧であったことが原因であると結論付けられている．したがって本研究では，音響の立体感および視聴者と AR オブジェクトの位置関係を考慮し，4 項目を通して音響の評価を行った．設問 5，6 ではユーザインタフェースの評価としてそれぞれ音響の ON/OFF 操作，十字コント

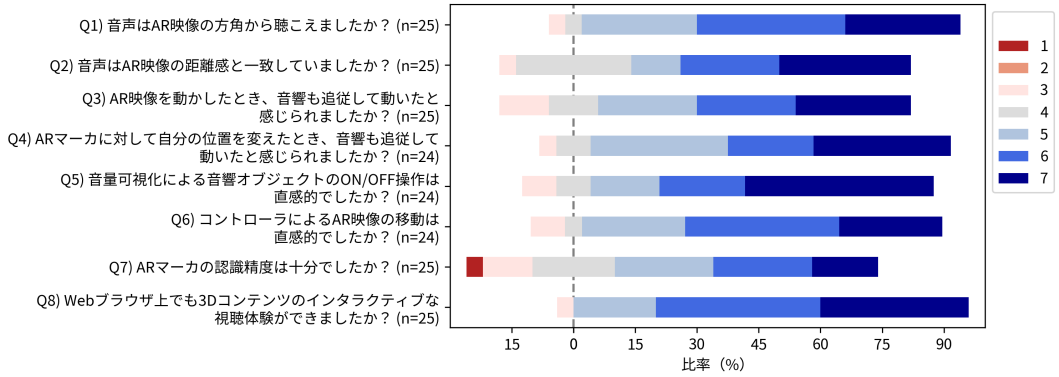

図 9　主観評価の集計結果

ローラによる AR 映像の移動操作に対する評価を問う．設問 7 では AR.js によるマーカ検出の精度を問い，設問 8 では総合的に Web ブラウザを用いた co-Sound による視聴体験の評価付けを行った．

評価結果

主観評価の結果を図 9 に示す．縦軸は前項で述べた Q1 – Q8 までの質問項目およびその有効回答数を，横軸は 1 – 7 までの 7 段階評価の回答比率を積み上げ棒グラフで表し，中間評価を表す 4 の回答比率の中間を原点に置いた．評価 5，6，7 が多いほど横軸正方向に偏り，評価 1，2，3 は負方向に偏る．

設問 Q7 を除いた全ての項目において，評価 6 および評価 7 を合わせた回答比率が 50%を上回り，総合評価である Q8 においては 76%であった．評価 3 以下の合計回答比率は，設問 Q3 が 12%，Q7 が 16%であったが，その他設問は全ては 10%未満となった．一方，設問 Q7 は AR.js のマーカ推定精度を問うものであったが，平均評価は 4.96，最高評価の評価 7 の回答比率は 16%，更に最低評価である評価 1 の回答も存在した．全項目中で平均評価が 5 未満となったのは設問 Q7 のみであり，評価 7 の回答比率も最低であった．

6. 考察

6.1 性能評価の考察

実験 1–4 より，提案手法で用いた WebRTC による端末間通信は，WebSocket による通信と比較して適当であると言える．

第一に，パケット転送のリアルタイム性が挙げられる．空間共有型 AR における QoE (Quality of Experience) についての評価は未だ定まっていないが，西堀らのインターネットを介した音楽セッションの遅延認知に関する研究 [12] では，30 ms 以上で遅延を認知し，50 ms 以上では演奏が困難になると報告されており，また一方で VR での FPS (First Person Shooting) ゲームでは，100 ms 以上でプレイヤーのスコアおよび QoE (Quality of Experience) が低下すると示されている [13]．本実験の結果より，WebSocket を用いた通信では遅延が 100 ms 以上であるが，WebRTC を用いた通信では平均遅延が 50 ms 以下である．同一 LAN 内での P2P はワンホップで通信が可能であり，加えて WebRTC は SCTP プロトコルを用いることでオーバーヘッドを低減し，HTTP プロトコルを用いた通信よりも低遅延性を保つことが可能であることが確認できた．

第二に，実験で測定した範囲内で転送遅延はメッセージサイズ・接続端末数に非依存であったことが挙げられる．オブジェクト数の増加や属性の複雑化により送信パケットのペイロードが長くなった場合でもリアルタイム性に問題がなく拡張性が高いと言える．また一般的にサーバ–クライアント間でコネクションを確立する WebSocket と異なり，WebRTC ではサーバレスな P2P 通信方式，すなわち接続端末数分のコネクションを確立する必要があるが，10 台以下の小規模であればコネクションの増加に伴う CPU 負荷は遅延に影響しないと結論付けられる．

6.2 アンケート評価の考察

はじめに，音響に関する設問である Q1 – Q4 について述べる．Q1 および Q4 の評価 5 – 7 の合計回答比率はそれぞれ 92.0%，87.5%となった一方で，Q2 および Q3 は評価 5 – 7 の合計回答比率が 68%，76%と前者の設問に比べて大きく評価が下がった．Q1 および Q4 は AR 映像に対する音像の方向・位置を問うものであった．Q2 および Q3 に関しても音響についての設問であったが，前者と異なり AR 映像に対する音像の距離感を問うたものであった．すなわち本研究で試作したアプリケーションにおいて AR 映像に対する音像の方位追従性は優れているが，音像の距離追従性が悪いと言え，加藤らによる先行研究 [5] において音響の立体感に対する評価が分散した原因に音像の正しい距離感があると結論付けられる．方位追従性に比べ距離追従性

の評価が低いという結果は，WebAudio PannerNode で用いられているバイノーラルアルゴリズムが簡易なものであるため，実空間とのキャリブレーションが不十分であることを示唆している．頭部伝達関数を考慮した HRTF モードも存在するが，CPU 負荷が高く，WebAudio の仕様上，方位角と仰角に依存する遅延が加えられることから本研究におけるリアルタイムレンダリングの要求を満たせないため不適当である．

ユーザインタフェースに関する設問 Q5,6 は，どちらも LiVRation，Web360^2 と同等に高い評価を受けた．

次に AR.js に関する設問 Q7 について述べる．前節で述べた通り，他の項目に比べマーカの検出精度は低い評価であった．AR.js で利用される ARToolkit はマーカ検出に原始的なアルゴリズムを用いており，偽陰性率が高い [14] ことで知られているが，本研究においても主観評価に現れる結果となった．WebAR はブラウザベースのものが主であるが，視聴者にとって高い評価を得るには精度が不十分であることが結論付けられる．

最後に，自由記述項目について述べる．「立体的な音でおどろいた」等肯定的意見は音響に関するものであった一方で，否定的意見はマーカの認識精度，ユーザインタフェースの二つに大別された．前者は Q7 にも反映されていた通りである．後者は「オブジェクトをタップしても反応しないことがあった」「音響オブジェクトは指で動かしたい」等，AR オブジェクトの動かし方が直感的でない・精度が悪いという意見も多かった．設問 Q5 および Q6 の結果よりタッチによる音響操作はインタラクティブ性が高いと言えるため，AR オブジェクトの移動操作 UI には改善の余地がある．

7. 結論および今後の課題

本研究では，AR，特に WebAR を利用したインタラクティブな視聴を可能とする映像音声メディアを提案し，Web アプリケーション「co-Sound」を実装した．AR を活用し，視聴者からのオブジェクト操作を動的に映像音声と組み合わせたマルチモーダル・インターフェイスを設計することで，実空間との親和性が高い映像音声のディジタル空間の合成提示およびインタラクティブなコンテンツ視聴が可能となった．さらに視聴メディア端末間の低遅延双方向通信によって，視聴者間同士がコンテンツの送受信者となり，ユーザ同士のインタラクションを実現することができた．

co-Sound の課題として，第一に AR オブジェクトの永続的な状態管理が挙げられる．現在の co-Sound では，より低遅延な双方向通信を実現する WebRTC を利用しているが，サーバーレスな P2P 通信である．同一 LAN 内ではワンホップでの通信が可能となるため有意であるが，同時間帯に存在するメッシュネットワーク内ピア間でしかコンテンツ情報が保持されないという問題点がある．さらに端末間コネクションを各ピアが独自に確立し，複数のピアでほぼ同時刻に送出されたオブジェクト操作情報のレンダリングはパケットの到達順に依存するため，空間情報が一意でなくなる可能性がある．タイムスタンプによる各ピアでのシーケンス処理は，高精度な時刻同期をブラウザ上で実現することが要求され実装は困難である．以上の理由から外部サーバによる永続的な状態管理システムが必要である．第二に，WebAR の性能限界が挙げられる．本研究で行った主観評価の結果にあるように，WebAR のマーカ検出精度は視聴者の QoE を下げる．ブラウザカーネルベース AR やアプリケーションベース AR であっても使用可能なリソースは限られることもあり，近年では AR にモバイルエッジコンピューティングを活用する研究も行われている [15]．これを WebAR にも応用し，現在モバイルで行っている AR 演算処理を外部リソースに分散し，高精度なトラッキング・リアルタイムなレンダリングが期待できるが，これは今後の課題とする．

参考文献

[1] 2023 年までの世界 ar/vr 関連市場予測を発表. https://www.idc.com/getdoc.jsp?containerId=prJPJ45301519. (Accessed on 05/01/2020).

[2] Sevda Küçük, Samet Kapakin, and Yüksel Göktaş. Learning anatomy via mobile augmented reality: Effects on achievement and cognitive load. *Anatomical Sciences Education*, Vol. 9, No. 5, pp. 411–421, 9 2016.

[3] Manabu Tsukada, Keiko Ogawa, Masahiro Ikeda, Takuro Sone, Kenta Niwa, Shoichiro Saito, Takashi Kasuya, Hideki Sunahara, and Hiroshi Esaki. Software Defined Media: Virtualization of Audio-Visual Services. In *IEEE International Conference on Communications (ICC2017)*, pp. 1–7, Paris, France, 2017.

[4] Takashi Kasuya, Manabu Tsukada, Yu Komohara, Shigeki Takasaka, Takuhiro Mizuno, Yoshitaka Nomura, Yuta Ueda, and Hiroshi Esaki. Livration: Remote vr live platform with interactive 3d audio-visual service. In *IEEE Games Entertainment & Media Conference (IEEE GEM) 2019*, pp. 1–7, Yale University, New Haven, CT, U.S., 2019.

[5] Shin Kato, Tomohiro Ikeda, Mitsuaki Kawamorita, Manabu Tsukada, and Hiroshi Esaki. Web360^2: An Interactive Web Application for viewing 3D Audio-visual Contents. In *17th Sound and Music Computing Conference (SMC)*, Torino, Italy, 2020.

[6] Cristina Fenu and Fabio Pittarello. Svevo tour: The design and the experimentation of an augmented reality application for engaging visitors of a literary museum. *International Journal of Human-Computer Studies*, Vol. 114, pp. 20 – 35, 2018. Advanced User Interfaces for Cultural Heritage.

[7] A. B. Tillon, I. Marchal, and P. Houlier. Mobile augmented reality in the museum: Can a lace-like technology take you closer to works of art? In *2011 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities*, pp. 41–47, Oct 2011.

[8] 塚田学, 菰原裕, 粕谷貴司, 新居英明, 高坂茂樹, 小川景子, 江崎浩. SDM360^2：インタラクティブ 3D コンテンツの

自由視聴点再生. 情報処理学会論文誌デジタルコンテンツ（DCON）, Vol. 6, No. 2, pp. 10–23, aug 2018.

[9] Ray Atarashi, Takuro Sone, Yu Komohara, Manabu Tsukada, Takashi Kasuya, Hiraku Okumura, Masahiro Ikeda, and Hiroshi Esaki. The Software Defined Media Ontology for Music Events. In *Workshop on Semantic Applications for Audio and Music (SAAM) held in conjunction with ISWC 2018*, pp. 15–23, Monterey, California, USA., 2018.

[10] Webrtc 1.0: Real-time communication between browsers. https://www.w3.org/TR/webrtc/. (Accessed on 05/01/2020).

[11] Iván Santos-González, Alexandra Rivero-García, Tomás González-Barroso, Jezabel Molina-Gil, and Pino Caballero-Gil. Real-time streaming: A comparative study between rtsp and webrtc. In Carmelo R. García, Pino Caballero-Gil, Mike Burmester, and Alexis Quesada-Arencibia, editors, *Ubiquitous Computing and Ambient Intelligence*, pp. 313–325, Cham, 2016. Springer International Publishing.

[12] 西堀佑, 多田幸生, 曽根卓朗. 遅延のある演奏系での遅延の認知に関する実験とその考察. 情報処理学会研究報告. [音楽情報科学], Vol. 53, pp. 37–42, dec 2003.

[13] S. Vlahovic, M. Suznjevic, and L. Skorin-Kapov. Challenges in assessing network latency impact on qoe and in-game performance in vr first person shooter games. In *2019 15th International Conference on Telecommunications (ConTEL)*, pp. 1–8, July 2019.

[14] M. Fiala. Artag, a fiducial marker system using digital techniques. Vol. 2, pp. 590 – 596 vol. 2, 07 2005.

[15] Ali Al-Shuwaili and Osvaldo Simeone. Energy-Efficient Resource Allocation for Mobile Edge Computing-Based Augmented Reality Applications. *IEEE Wireless Communications Letters*, Vol. 6, No. 3, pp. 398–401, 1 2017.