サイバースペースと仮想都市研究会 CSVC 2023 40
複合現実感研究会 MR2023 21
2023.10.26・27)

実世界観測による時空間映像データの高度利用 (4)

—時空間映像データ配信における C/S 方式と P2P 方式の比較—

江崎 佑真*1　西川 卓*1　中村 文彦*1　木村 朝子*1　柴田 史久*1

Abstract --- 我々はこれまで，カメラ等から取得されたセンシングデータを収集，蓄積，活用するための枠組みである SIGMA フレームワークを開発してきた．これまでの SIGMA フレームワークでは，センサ端末とアプリケーション端末が同一インターネット内に存在する場合であってもサーバを介して通信する必要があった．しかし，リアルタイム性の高いアプリケーションを開発する場合，より低遅延での配信を実現する必要がある．そこで本稿では，新たに P2P 方式でストリーミング配信しつつデータベースにデータの蓄積をする蓄積配信機構を新たに実装し，センサ端末が観測してからアプリケーション端末へフレームが届くまでの配信遅延を C/S 方式での配信と比較を行った．目標遅延を定めて結果を比較することで，開発者がアプリケーション開発で配信方式を決定する時に各アプリケーションに適切な配信方式を示すガイドラインを作成した．

Keywords: 時空間映像データ，メタデータ，低遅延配信，ストリーミング配信

1 はじめに

実世界を観測するセンサが急速に普及しており，街中に遍在するようになってきた．例えば，車載カメラや街頭に設置される定点カメラ，ドローン搭載カメラなどである．自動運転やドローン配達などが今後普及することが予想されるため，それに伴いセンサの数は増加していくであろう（**図 1**）．このような背景から我々は，街中に遍在するセンサ機器が取得したセンシングデータを収集，蓄積し，それを活用するためのアプリケーションフレームワーク SIGMA (Spatiotemporal Images with Generalized Management Architecture) を開発してきた[1-4]．

SIGMA フレームワークが目指すのは，実世界を観測した時間的・空間的に遍在する画像やセンサデータを獲得・伝送・蓄積・変換・加工・表示するための仕組みを実現することである（**図 2**）．様々な映像データ・センサデータを獲得し，それを伝送・加工した上で，アプリケーションで利用するとともに，蓄積や変換をした上で再利用するような循環型のフレームワークを目指している．SIGMA フレームワークを利用して実現可能なアプリケーションとしては，例えば，歴史的建造物の 3 次元再構成や 3 次元地図の作成など蓄積された大量のデータを利用するものや，街頭の定点カメラや街中を飛行しているドローンの画像を使用した防犯用の監視システム，複数の車載カメラ LiDAR の画像データと点群データを利用した ITS(Intelligence Transport System) 分野のアプリケーションなどリアルタイムのセンサデータを活用するものなどである．

SIGMA フレームワークの利用を想定するアプリケーションの中には低遅延配信を必要とするものがある．その一例として，自動車の死角を隠消現実感 (DR; Diminished Reality)処理によって可視化する

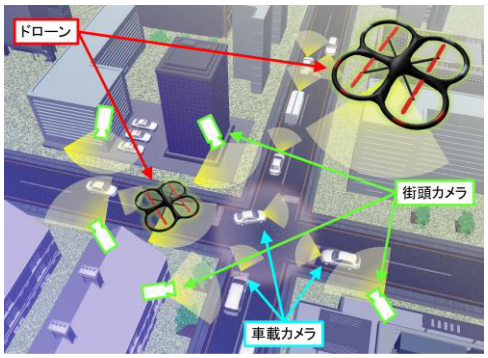

図 1 街中に遍在するセンサ機器の例

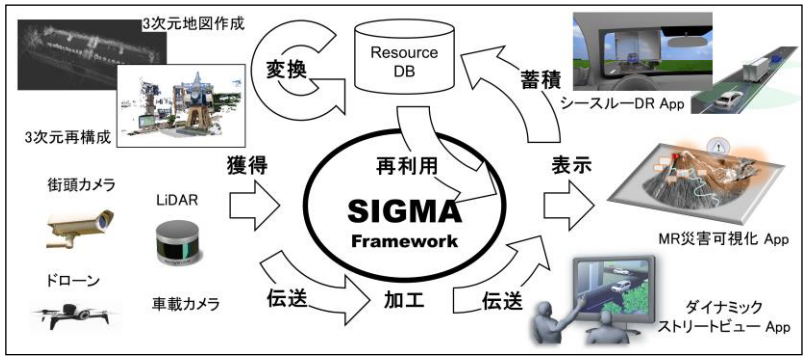

図 2 SIGMA フレームワークの概念図

*1 立命館大学大学院 情報理工学研究科

アプリケーション（シースルーDR アプリケーション）について考える[5-9]．DR 処理のために必要な画像データは，周囲のセンサから SIGMA フレームワークを介して配信される．この時，配信データに遅れが生じると DR 処理後に運転者の死角領域を可視化できない可能性があり，これにより，自動車事故が起こりうる．SIGMA フレームワークの過去の研究では，低遅延なデータ配信を実現するためにリアルタイムなデータ配信を行うストリーミングサーバを利用した C/S（Client-Server）方式による配信機構の設計・実装に取り組んだ[4]．しかし，この機構では同一ネットワーク内にセンサ端末とアプリケーション端末がある場合も外部のサーバを経由する必要がある．同一ネットワーク内で配信を行う際には端末同士が直接配信する P2P（Peer to Peer）方式を用いることでより低遅延な配信が期待される．

そこで本稿では，SIGMA フレームワークの配信機構に新たに P2P 方式を用いた通信機能を設計・実装する．また，C/S 方式と P2P 方式の配信遅延を比較し，開発者はどのような基準でこれらを使い分けるべきか考察する．

2 SIGMA フレームワーク

2.1 SIGMA フレームワークの概要

SIGMA フレームワークは，センサデータを収集，蓄積すると同時に，それを利用したアプリケーションを作成するためのアプリケーションフレームワークである．

SIGMA フレームワークでは，様々な場所，時間で取得したセンサデータを「時空間映像データ」と呼んでいる．時空間映像データは，センサの位置や姿勢，撮影時刻などのメタデータが関連付けられている．時空間映像データとしてサーバで蓄積・管理することにより，後から 3 次元再構成での利用や 3 次元地図の作成が可能になる．

2.2 従来の蓄積配信機構と課題

従来の SIGMA フレームワークは 4 つのサーバで構成されている[4]．クライアントとやり取りおこなう Web サーバ，データを保存するデータベースサーバ，クライアントからのリクエストに応じて検索，加工を行うアプリケーションサーバ，受信したデータを中継するストリーミングサーバの 4 つである．ストリーミングサーバでは，センサから送られてきたデータをクライアントで実行されるアプリケーションまで中継するとともに，データベースにデータを保存することが可能であり，ある程度リアルタイム性のあるアプリケーションを実行可能である．しかしこの構成では，サーバを必ず経由する必要があるため，アプリケーションにとって都合が悪い場合が考えられる．例えば，センサ端末とアプリケーション端末が同一ネットワーク内に存在するアプリケーションを考えた場合，既存の配信機構でストリーミング配信を行おうとすると，サーバを経由するために一度外部ネットワークにデータを送る必要がある．サーバが端末から離れた場所にある場合，データの転送数（ルータのホップ数）が増加し，遅延が大きくなる．

そこで本研究では，同一ネットワーク内にセンサ端末とアプリケーション端末が存在する場合に，両者間の通信がより低遅延で行えるように新たに P2P 方式を SIGMA フレームワークに導入する．SIGMA フレームワークを利用するアプリケーションによっては C/S 方式が望ましいもの，P2P 方式が望ましいもの，そしてどちらでも問題ないものが想定される．そこで，C/S 方式と P2P 方式の遅延を比較することで，開発者が配信方式を決定する際に，参考にできるガイドラインを作成する．

3 P2P 方式を用いる新たな蓄積配信機構

3.1 関連研究

Google などが中心となって開発された WebRTC[10]では，カメラで撮影された映像や音声などのリアルタイムなメディアデータを低遅延でストリーミング配信することが可能である．ストリーミング配信は，メディアデータを細かく分割してからインターネットなどのコンピュータネットワーク上で送信することで配信する手法で，データ全体をダウンロードしなくても利用できるという特徴を持つ．WebRTC では RTP（Real-time Transport Protocol）を使ったメディアデータ配信に加え，SCTP（Stream Control Transmission Protocol）を使用したメディアデータ以外のデータ配信も可能であり，特に SCTP を使ったデータ配信方法はデータチャネルと呼ばれる．データチャネルはテキストメッセージやファイルの送受信に使われることが多く，いかなる形式のデータも扱うことも可能である．

Jaouhari らは手術室内の内視鏡・超音波映像などと他の医療用センサで取得したメタデータを外部に低遅延でストリーミング配信する手法を提案している[11]．Jaouhari らの提案手法は，手術用器具で取得される動画データとメタデータを WebRTC の RTP を使ったメディアデータ配信とデータチャネルにより P2P 方式で手術室外部に配信する．Jaouhari らは提案手法を実利用する上で課題となる点を 3 つ挙げており，その中の 1 つとして動画データとメタデータの同期が難しいという点がある．SIGMA フレームワークの場合でも，配信するセン

サデータには様々な場所・時間で取得されたことを表すメタデータが付与されているため，これらを同期して配信できないことは大きな問題となる．

Amirante らは，P2P 方式での通信を前提としている WebRTC を C/S 方式での通信でも利用可能にするための Janus WebRTC Server を開発している[12-14]. Janus サーバでは全てのクライアントとの間で WebRTC ピア接続を作成し，カメラなどのセンサが接続されたクライアントとのピア接続によってデータを受信し，他のクライアントとのピア接続へデータを中継配信することで C/S 方式での通信を実現している．また，Janus サーバは汎用性の高い WebRTC サーバとして開発されており，利用者が必要とする機能をプラグインとして追加し利用する方式を採っているため，必要最小限のリソースのみ消費されるという特徴もある．

Janus WebRTC Server では標準で録画機能が利用可能で，Janus サーバで中継配信する動画データと音声データを Meetecho Janus Recordings (mjr) 形式で保存することが可能である．この録画機能は Janus サーバでクライアントから受信したデータ（RTP パケット）をそのまま保存するもので，動画データ・音声データとして使用するためには変換処理が必要となる．そのため，SIGMA フレームワークのように連続してデータを検索可能な形でデータベースなどに蓄積することは難しい

これらを踏まえて本稿では，SIGMA フレームワークに WebRTC をベースとした P2P 方式が利用可能な蓄積配信機構を新たに追加する．また，通信方式として P2P 方式に加えて C/S 方式も用意することで，両者を比較できるようにする．

3.2 新たな蓄積配信機構の設計

本機構で提供するストリーミング配信は，Amirante らの Janus WebRTC Server をベースにし，SIGMA フレームワークで必要な機能を追加することで実現する．センサピアはサーバを介することなくアプリケーションピア（アプリピア）に直接配信するとともに，サーバへデータのアップロードも行う(**図 3**)．WebRTC のデータチャネルでは任意の形式のデータを送受信できるため，カメラで撮影されたばかりの画像データに様々なメタデータを埋め込み，アプリケーションに配信することができる．また，SIGMA フレームワークのアプリケーションを使用するすべての端末に Janus WebRTC Server をインストールすることで，P2P 配信を実現する．Jaouhari らが課題点として挙げたように，WebRTC はストリーミング配信するデータとメタデータを同期する機能を持たない．そのため，本機構ではセンサ側の端末が動画データにメタデータを埋め込むことで同期し，アプリケーション側の端末でメタデータを抽出，アプリケーションで利用できるようにする．なお，メタデータが埋め込まれた動画データは SCTP を使用したデータチャネルで配信する．3.1 節で述べたようにデータチャネルでは任意のデータ配信が可能で，メタデータが埋め込まれた動画データの配信も可能である．

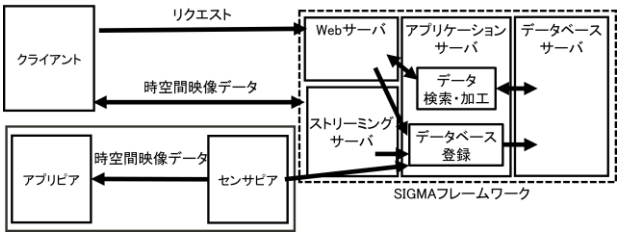

図 3 ピアを含めたサーバ構成

他の実装方法としては，RTP を使用したメディアデータ配信方法において Encoded Transform API[15]を使用することでメタデータを埋め込むことが可能であるが，RTP を使用したデータ配信では帯域幅などの条件により動画データの解像度が WebRTC に変更されることがある．解像度が小さくなるとオリジナルのデータから離れたデータとなってしまうため，動的に変更されることは望ましくない．そのため，本提案機構では Janus WebRTC Server でデータチャネルを使用するためのプラグインをベースとして実装する．

3.3 新たな蓄積配信機構の実装

本機構ではストリーミング配信と従来のデータ配信で受け取ったデータを一様に扱えるようにするために，動画データを画像データに変換してアプリケーションに受け渡す．データの形式についても従来の機構で利用していた PNG に統一することで，従来同様の処理が適用可能になる．ストリーミング配信は Web ブラウザ上のデータをアプリケーションに受け渡す必要がある．そのため，本機構ではリアルタイムなデータの受け渡しが可能であり，センサデータとメタデータを組み合わせて配信できる ROS[16]の Topic 通信[17]を利用する．

C/S 方式と P2P 方式では配信の流れが異なるため，データを蓄積する端末とタイミングが異なる．C/S 方式ではサーバがデータの蓄積を行う．これはサーバ内部で実行する ROS アプリケーションであるデータ蓄積アプリケーションを用いて実現する．データ蓄積アプリケーションはサーバが中継配信するデータを内部で受信し，画像に変換後，データベースへ登録する．

一方，P2P 方式のストリーミング配信はピア同士で通信するため，ストリーミングサーバの代わりに

センサピアが蓄積処理を行う．そのため，センサピアは，アプリケーションピアへのデータ配信とデータ蓄積の 2 つの処理をする必要があり，負荷が大きくなる．P2P 方式の蓄積処理は C/S 方式の蓄積において用いたデータ蓄積アプリケーションをセンサピア内部で実行することで実現する．

4 実験

4.1 実験

C/S 方式と P2P 方式の配信遅延を計測し，目標遅延と比較することで SIGMA フレームワークにおける配信方式のガイドラインを作成する．実験では 5 分間ストリーミング配信を行い，遅延の計測を行う．各フレームの目標遅延として今回は以下の二つを設定する．なお，以下の遅延は映像がカメラで撮影されてから配信後にアプリケーション上で表示されるまでの時間である．

目標① 150ms 未満が望ましく，400ms を超過するべきではない

目標② 300ms を超過するべきではない

すなわち，実験では各配信方式で 150ms 未満の遅延を目指しつつ，300ms または 400ms を超過せずにデータ配信することが目標になる．これらの目標遅延を SIGMA フレームワークで想定される使用用途を参考に設定することで，各配信方式を実際に利用する際に配信遅延で問題が起こるかどうか考察する．目標①はコミュニケーション用途を想定し，通話アプリやリモート会議アプリなどが例として挙げられる．この目標は ITU-T の G.114 勧告[18]に基づいて設定した．目標②は ITS 分野のアプリケーションを想定し，走行中の自動車を画像認識する際に発生する誤差を車両 1 台分以下に抑えられる遅延を基に設定した．認識する自動車は時速 60km で全長 5m と想定している．

実験では遅延として Round-Trip Time (RTT)を計測する．RTT とは通信相手に送信したデータが返送されて受信完了するまでの時間である．なお，実験では動画データに埋め込まれているメタデータの時刻情報を利用する．また，実際の利用環境に近づけて遅延を計測するため，C/S 方式と P2P 方式で配信に利用する端末やその位置が異なる．C/S 方式ではサーバを利用する配信方式であり，クライアントとは異なるネットワークに位置するサーバと通信することを想定している．一方，P2P 方式はセンサピアとアプリケーションピアが同一ネットワークに存在することを想定する．そのため，インターネットを介する C/S 方式は，より遅延が大きくなることが予想される．また，配信方式のガイドラインを作成するために，動画データの解像度についても 2 つを条件に加えた．したがって，実験条件は以下の 2 種類の条件の組み合わせにより，合計 4 条件になる．

1. 通信方式
   (a) P2P 方式
   (b) C/S 方式
2. 解像度
   (a) 1280×720 (HD; High-Definition)
   (b) 1920×1080(Full-HD)

実験で使用した遅延計測用クライアントとストリーミングサーバの仕様を**表 1** に，遅延計測用ピアとストリーミングピアの仕様を**表 2** 示す．なお，本実験ではストリーミングサーバとして GMO が提供する ConoHa VPS(Virtual Private Server)を使用した[19].

4.2 計測結果

実験によって得られた配信遅延を**図 4～7** に示す．計測した各方式の遅延を**表 3** に示す．表 3 より HD 画質での配信の平均遅延はネットワーク接続方法に問わず，目標①の下回ることが望ましい遅延よりも小さい．また，図 4，図 5 から Full-HD 画質の配信では P2P であれば平均遅延が 300ms を下回るものの超えたフレームも存在し，C/S 方式では平均遅延が 300ms を超え，400ms 以上の遅延が生じたフレームが存在した．

4.3 配信方式のガイドライン

実験の結果からリアルタイムなデータを利用するアプリケーションを開発する際のガイドラインを検討する．

表 3 より，HD 解像度のデータであればいずれの配信方式でも平均遅延が「150ms を超過するべきではない」を達成しており，300ms を超過するものがないのでコミュニケーション用途や ITS 用途で利用する際は通信方式を自由に選択できる．ただし，C/S 方式の分散が大きいことから安定性を考慮すると P2P のほうがより望ましい．また，Full-HD 解像度のデータの配信遅延のうち，P2P 方式は目標①の「150ms を超過するべきではない」を達成することができなかったが，目標②の「300ms 以下が望ましい」を達成した．一方，C/S 方式は平均遅延が目標②の「300ms 以下が望ましい」を達成できず，またフレームによっては，目標①の「400ms を超過するべきではない」を達成できていない．そのため，

表 1 実験で使用したクライアントとサーバ

	遅延計測用クライアント	ストリーミングサーバ
OS	Ubuntu 18.04	Ubuntu Server 20.04
CPU	Intel Corei7-7700HQ8コア	Intel Xeon Gold 6230 6コア
メモリ	32GB	8GB
GPU	NVIDIA GeForce GTX 1050 Ti	
その他		VPS(Virtual Private Server)を使用

表 2 実験で使用したピアのスペック

	遅延計測用ピア	ストリーミングピア
OS	Ubuntu 18.04	Ubuntu 20.04
CPU	Intel Core i7-7700HQ8コア	Intel Core i5-8400 6コア
メモリ	32GB	32GB
GPU	NVIDIA Deforce GTX 1050Ti	NVIDIA GeForce RTX 3060

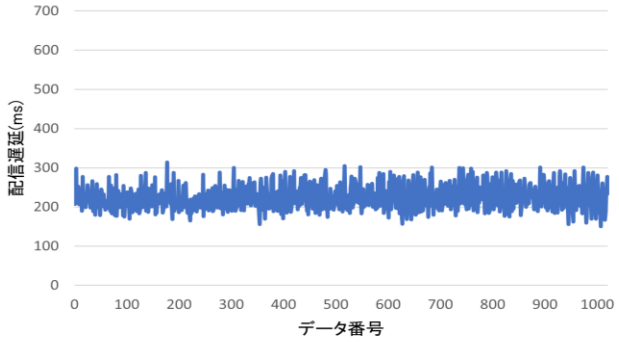

図 4 P2P 方式，Full-HD での遅延の推移

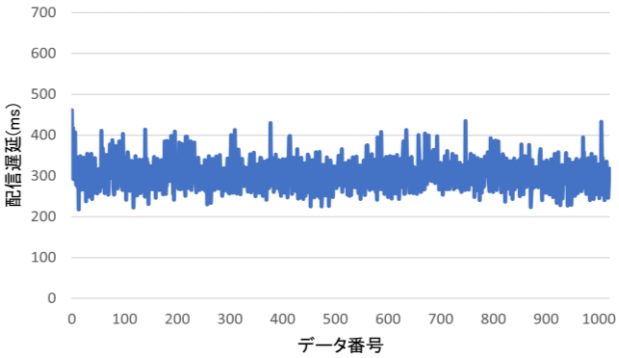

図 5 C/S 方式，Full-HD での遅延の推移

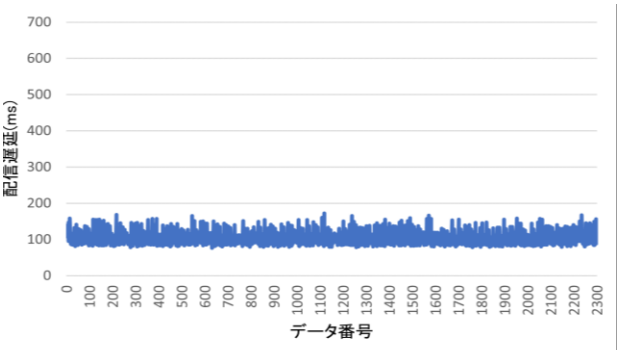

図 6 P2P 方式，HD での遅延の推移

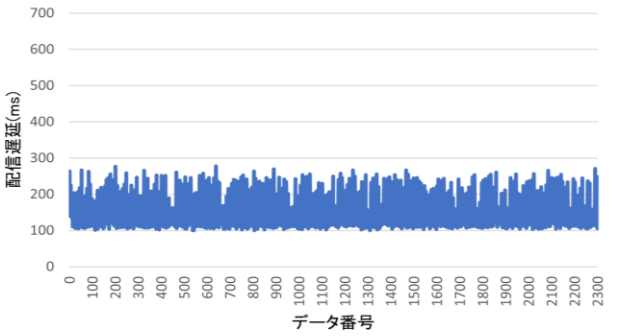

図 7 C/S 方式，HD での遅延の推移

Full-HD 解像度を利用する際はコミュニケーション用途において C/S 方式はユーザに違和感を与える可能性があり，P2P 方式でも 1.7%のフレームが 300ms を超過したため違和感を与える可能性がわずかながら存在する．また，ITS 用途において C/S は 1.6%のフレームが 400ms を超過したため，遅延による問題が発生することが考えられ，P2P 方式は 400ms を超過したフレームはないため，問題はないと考えられる．

表 3 実験で計測した遅延

	Full-HD		HD	
	P2P	C/S	P2P	C/S
平均(ms)	224.17	303.13	101.35	139.17
中央値(ms)	221	298	99	130
最大値(ms)	314	461	172	277
最小値(ms)	150	218	76	100
分散(ms^2)	848.267	1310.062	238.555	1115.078
150以上%	100	100	1.7	18.96
300以上%	0.78	48.63	0	0
400以上%	0	1.57	0	0

以上をまとめると，SIGMA フレームワークにおいてリアルタイムなデータを使用するアプリケーションを開発する際に参照すべきガイドラインは，図 8 に示すフローチャートのようになる．これを参考にすることで開発者は，通信方式やネットワークへの接続方法を決定できる．フローチャートでは ITS 用途のアプリケーションが通信方式として優先的に P2P 方式を選択すべきで，一般的なコミュニケーション用途のアプリケーションではなるべく C/S 方式を選択すべきであるとしている．ただし，HD 解像度で利用する場合は配信方式による遅延の影響は小さいのでアプリケーションのほかの条件に合わせて通信方式を選択できる．また，ITS 用途で Full-HD 解像度のデータを利用する場合は，P2P 方式を使用したほうが良い．なお，同一ネットワーク内にピアが存在しないなどの理由で P2P 方式が利用できない場合は，遅延による問題が生じる可能性を考慮したうえで C/S 方式を使用するべきである．

5 むすび

本稿では我々が提案する SIGMA フレームワークにおいてデータの蓄積と配信を行う新しい蓄積配信機構を開発した結果について述べた．本機構により従来利用されていた C/S 方式に加え P2P 方式での配

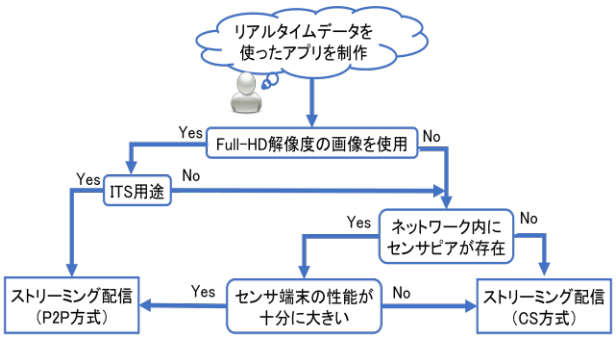

図 8 配信方法選択のフローチャート

信が可能になった．また，P2P 方式と C/S 方式で比較実験を行い通信方式選択時に利用可能なガイドラインを作成した．実験の結果，HD 解像度のデータ配信において，いずれの配信方法でも目標遅延よりも低遅延で配信できることを確認した．また，Full-HD 解像度のデータ配信では遅延により，P2P 方式のみが達成する目標があることが確認できた．今後の展望として，蓄積配信機構における開始処理と終了処理について詳細な設計と実装が必要であると考える．特に，ストリーミング配信で必要となるセンサクライアントとセンサピアの検索方法について検討が必要であると考える．アプリケーションが利用する配信方法が C/S 方式かP2P方式かという条件に応じた検索方法が求められる．

謝辞

本研究の一部は，科研費：基盤研究（B）課題番号 21H03487 による

参考文献

[1] 山﨑賢人, 有富友紀, Guan Sikun, 木村朝子, 柴田史久: 実世界観測による時空間映像データの高度利用 (1) ―基本アーキテクチャの概念設計と第 1 次システム試作―; 日本 VR 学会研究報告, Vol. 22, No. 2, pp. 49 - 53 (2019)

[2] 山﨑賢人, Guan Sikun, 松木輝, 木村朝子, 柴田史久: SIGMA フレームワークにおける時空間映像データの管理手法; 情報処理学会研究報告デジタルコンテンツクリエーション(DCC), Vol. 2020-DCC-26, No. 9, pp. 1 - 6 (2020)

[3] 山﨑賢人, Guan Sikun, 松木輝, 木村朝子, 柴田史久: SIGMA Retriever: イメージベースドモデリング・レンダリングのための検索機構の設計と実装; 情報処理学会論文誌 デジタルコンテンツ, Vol. 10, No. 1, pp. 16 - 27 (2022)

[4] 西川卓, 松木輝, 山﨑賢人, 木村朝子, 柴田史久: 実世界観測による時空間映像データの高度利用 (3) ―時空間映像データの低遅延な蓄積配信機構の設計と実装―; 日本バーチャルリアリティ学会複合現実感研究会, MR2022-8, Vol.25, No.1 (2022)

[5] 平松黎, 有富友紀, 若林優, 木村朝子, 柴田史久: 安全運転支援のための周辺車両の半隠消表示法 (1) ― 車車間通信モジュールの設計と実装 ―; 第 25 回日本バーチャルリアリティ学会大会論文集, 3B1-3 (2020)

[6] 若林優, 竹村岩朗, 平松黎, 木村朝子, 柴田史久: 安全運転支援のための周辺車両の半隠消表示法 (2) ― 移動物体を含む死角領域の可視化 ―; 同上, 3B1-4 (2020)

[7] 平松黎, 若林優, 佐々木俊希, 木村朝子, 柴田史久: 安全運転支援のための周辺車両の半隠消表示法(3) ― 5G 環境下における性能評価実験―; マルチメディア, 分散, 協調とモバイル (DICOMO2021) シンポジウム, pp. 980 - 986 (2021.7.1)

[8] 藤重秀斗, 若林優, 松室美紀, 木村朝子, 柴田史久: 安全運転支援のための周辺車両の半隠消表示法 (4) ― 半隠消表示法の拡張と評価 ―; 電子情報通信学会 ITS 研究会, Vol. 121, No. 436, ITS2021-63, pp. 1 - 6 (2022)

[9] 藤重秀斗, 松室美紀, 木村朝子, 柴田史久: 歩行者飛出し予測のための周辺車両の半隠消表示法, 日本バーチャルリアリティ学会論文誌, Vol. 28, No. 3, pp. 141 - 151 (2023) (in Press)

[10] Google“WebRTC”:https://webrtc.org/（2023 年 9 月 12 日）

[11] S. El Jaouhari, G Pasquier, A. Serrand, B. Gibaud, P. Hardy and E. Cordonnier: “ Streaming DICOM Real-Time Video and Metadata Flows Outside The Operating Room”, 2019 IEEE GlobalCommunications Conference (GLOBECOM), pp. 1 – 6, 2019.

[12] A. Amirante, T. Castaldi, L. Miniero, and S. P. Romano: “ Janus: a general purpose WebRTC gateway ”, In Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm '14), Article 7, pp 1 – 8, 2014.

[13] A. Amirante, T. Castaldi, L. Miniero, and S. P. Romano: “Performance analysis of the Janus WebRTC gateway”, In Proceedings of the 1st Workshop on All-Web Real-Time Systems (AWeS'15), Association for Computing Machinery, Article 4, pp. 1 – 7, 2015.

[14] A. Amirante, T. Castaldi, L. Miniero and P. Saviano: "Empowering Remote Participation in IETF Meetings through WebRTC", in IEEE Communications Standards Magazine, Vol. 1, No. 2, pp. 60 –66, 2017.

[15] World Wide Web Consortium (W3C)“WebRTC Encoded Transform”: https://www.w3.org/TR/webrtc-encoded-transform/（2023 年 9 月 12 日）

[16] ROS“ROS: Home”: https://www.ros.org/（2023 年 9 月 12 日）

[17] ROS Wiki“Topics”: http://wiki.ros.org/ja/Topics（2023 年 9 月 12 日）

[18] ITU-T,“Recommendation G.114: One-way transmission time”: https://www.itu.int/rec/T-REC-G.114-200305-I/en, 2003.

[19] GMO“ConoHa VPS”: https://www.conoha.jp/vps/（2023 年 9 月 12 日）