

# リアルタイム映像の収集と合成を伴う同世界放送システムの検討

牧田航輝<sup>†1</sup> 川上朋也<sup>†1</sup> 松本哲<sup>†2</sup> 義久智樹<sup>†2</sup> 寺西裕一<sup>†3,†2</sup> 下條真司<sup>†2</sup>

概要：2019 年末に発生した新型コロナウイルス感染症（COVID-19）の影響により、テレワークやオンライン授業など、リアルタイム映像配信サービスの需要が高まっている。複数の撮影対象を1つの画面上に表示する場合、画面を分割して各撮影対象を配置するのではなく、全撮影対象を1つの映像内に合成して配置することにより、より現実感のあるライブ放送が実現できる。本研究では、複数地点で撮影された対象が、まるで同じ空間に存在するかのようなライブ放送を「同世界放送」と呼び、そのシステムを検討する。同世界放送では、映像の合成処理が必要となるが、特定のノードに処理を集中させると遅延が大きくなるという問題があるため、中継ノードで合成しつつ収集し、最終的な合成映像を配信するノードまで転送する手法を提案する。LAN 環境での簡単な実装により、提案手法の効果を評価した。

## 1. はじめに

YouTube Live などのインターネットライブ放送サービスが、近年広く普及している。配信者は、このようなサービスを利用して映像や音声を配信する。そこでは、個人が単一の撮影対象のみを配信することだけでなく、複数の配信者によるコラボレーションもあり得る。また、2019 年の終わりに発生した新型コロナウイルス感染症（COVID-19）により外出自粛を求められ、テレワークやオンライン授業など、遠隔サービスが広まっている。多くの映像配信サービスでは、画面を分割することによって各配信者による撮影対象を配置する方法がとられている。しかし、それでは各撮影対象が孤立しており、配信者にとっても視聴者にとっても満足できる配信とは言い難い。配信者側では、複数の映像が分離されているために他の配信者の言動に対する反応が困難である。また、視聴者側では、通常のテレビ放送における中継のように遠隔地でのやり取りをしているような感覚となってしまふ。このように、双方にとってコラボレーションが行われている感覚が希薄となる問題がある。

画面を分割することなく、撮影対象を同じ空間に配置することができれば、高いエンターテインメント性を生み出したり、テレワークやオンライン授業などの際に生じ得る違和感や不快感を軽減したりする効果が期待できる。

本研究では、全ての撮影対象が同じ空間に存在するかのようなライブ放送を「同世界放送」と呼び、合成によって生成する映像を「同世界映像」と呼ぶ。このように、複数の配信者による撮影対象を一つの映像内に合成する方法も考えられる。そうすることで、全ての撮影対象がまるで同じ空間に存在するような、現実感のあるライブ放送が実現できる。同世界放送の適用例としてはまず、参加者が仮想空間内に自身や物体を配置した遠隔ミーティングなど、特定の対象の遠隔での存在感を高めることが考えられる。また、COVID-19 の影響で中止となった「仙台すずめ踊り」について、各演者が自宅で撮影した映像を募り 1 つの映像

に合成することで、仮想的にイベントを実現した例がある。この例は、録画された映像を編集したもので、リアルタイム映像の収集と合成ではないのだが、これをリアルタイムに合成することでさらなる現実感を生み出す可能性がある。このように、異なる場所にいる出演者のリアルタイム映像を用いた演劇や踊りなどのパフォーマンスでは、視聴者は出演者が同じ空間で共演しているかのような感覚が得られ、高い臨場感や現実感を同世界映像から得られる。

同世界放送で必要な映像合成の方法については、クロマキーや機械学習による被写体抽出技術<sup>2)</sup>を利用したデジタル合成などがある。しかし、同世界放送では、多数のリアルタイム映像をインターネット経由で収集し、低遅延に映像の合成と視聴者への配信を行う必要がある。そのため、既存の収集方式や処理割り当て方式を用いると、「撮影者（収集対象の映像）の増加に伴い、特定のノードに映像の収集と合成の処理の負荷が増大し、処理の完了までの遅延が大きくなる」という問題が生じ得る。この問題によってノード上の処理や通信の遅延が大きくなると、同世界映像が視聴端末で再生されるまでの待ち時間や視聴中の再生途切れ時間も増大する。それにより、インターネット放送サービス利用者の満足度が低下するため、インターネット放送サービス提供者にとって、快適なインターネット放送サービスを提供することは重要である。この問題を解決するため、分散コンピューティング環境における「逐次合成処理を伴う分散リアルタイム映像収集」を提案する。

以下、2 章で関連研究について説明し、3 章で本研究で提案する同世界放送システムについて述べ、4 章および 5 章で LAN 環境における実装とその評価を行う。6 章でシステムの紹介をし、最後に 7 章で本稿をまとめる。

## 2. 関連研究

リアルタイム映像の合成に関する類似研究がある。例えば、超高臨場感通信技術 Kirari<sup>3)4)</sup>では、あたかも目の前に別の空間にいる人がいるかのような超高臨場感を目指している。そのために、1 つの空間を複数カメラで撮影して得られた映像を合成することで、高い解像度と広い視野角を持つサラウンド映像を生成し、遠隔地へと伝送する技術を

<sup>†1</sup> 福井大学工学部電気電子情報工学科

<sup>†2</sup> 大阪大学サイバーメディアセンター

<sup>†3</sup> 国立研究開発法人情報通信研究機構

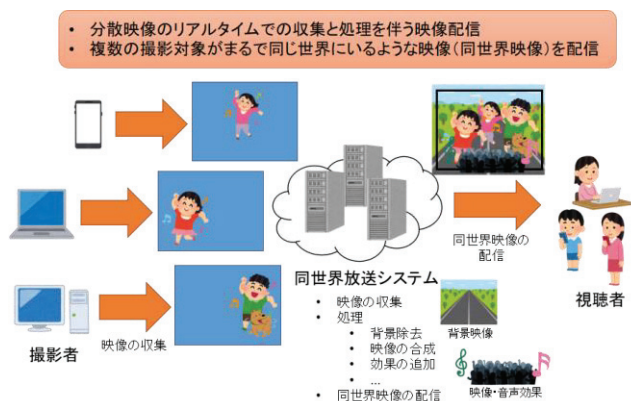


図1 同世界放送の構成

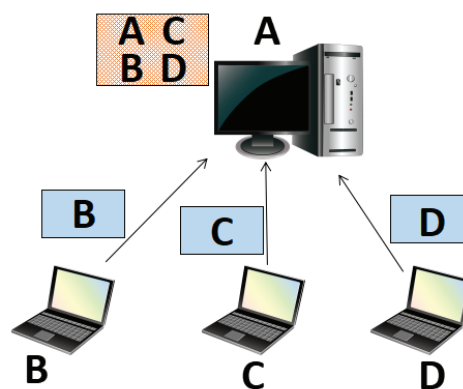


図2 集中合成

提案している<sup>5)6)</sup>。また、文献7)では、自由視点映像をリアルタイム配信するための映像合成技術と、5G網を用いた低遅延な配信システムを提案している。これらの研究は、ある特定の地点を複数カメラで撮影した映像を合成し伝送しているのだが、本研究のように複数地点で撮影された映像をインターネット経由で収集し合成することは想定されていない。また、複数カメラから得た映像を合成するのではなく、実写映像とCGを合成させる研究もある<sup>8)</sup>。ハンディカメラの動きを計測し、その動きをCG空間内の仮想カメラに反映させることによって実写映像とCGを合成させるためのハイブリッドセンサを開発している。

また、遠隔での存在感や臨場感に着目した類似研究もされている。文献9)では、異なる場所の参加者や家具をヘッドマウントディスプレイ(HMD)に表示し、まるで参加者や物体が付近に存在するかのように臨場感を高めている。文献10)では、ユーザと対話相手を同じ映像内に合成した時に、「同じ空間にいる」という感覚(ソーシャルテレプレゼンス)を強化させることを目的としている。文献11)では、カメラとプロジェクタを用いてリアルタイムに異なる部屋にいる人を実物大で映し出すことによって、存在感を高める技術を提案している。これらの研究はいずれも、遠隔での存在感や臨場感を高めることを目的としており、多数の映像の収集や合成、撮影対象者以外の視聴者への映像配信は想定されていない。

Microsoft TeamsのTogether mode<sup>12)</sup>では、参加者の上半身を切り抜き、最大49人までの参加人数に応じて各撮影対象を最適なサイズに変更し、共通の背景に自動的に配置される。そのため、表示位置を自由に設定できないという問題点はある。しかし、同じ空間内に参加者を表示することによって、より自然なコミュニケーションをとることが可能になるとされている。またAppleは、被写体が複数の自撮りを、全員がカメラの前に集まることなく撮影できるように、各々の端末で撮影した画像を合成させる特許を取得している<sup>13)</sup>。撮影した後も、各画像の位置を好きな位置に調整できるようになっている。

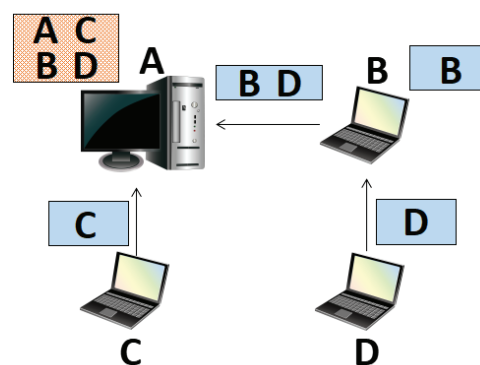


図3 分散合成

### 3. 同世界放送システム

#### 3.1 システムの構成と処理の流れ

本研究では、文献14)15)16)の異世界放送システムと同様に、複数の処理コンピュータがリアルタイム映像の収集や合成、視聴者へのライブ放送を行う分散型のモデルを想定している。同世界放送の構成を図1に示す。図1の例ではまず、同世界放送システムは合成元のリアルタイムな映像を撮影者からインターネット経由で収集する。同世界映像を生成するための処理には、元映像の背景除去(撮影対象の抽出)、複数映像の合成、映像・音声効果の追加などがある。映像合成時の被写体の位置合わせについては、ユーザ自身による映像内での移動や、重ね合わせの順序も含めて同世界放送システム側での何らかの規則に基づく調整を想定する。視聴者は、希望のライブ放送を同世界放送システムに指定し、生成された同世界映像の配信を受信する。

#### 3.2 映像の収集と合成における負荷分散

分散コンピューティング環境では、処理を複数台のコンピュータに分散させることで、1台あたりの処理負荷と通信負荷を軽減する。本研究の「逐次合成処理を伴う分散リアルタイム映像収集」では、「撮影者(収集対象の映像)が増えると、特定のノードに映像の収集と合成の処理が集中し、処理の完了までの遅延が大きくなる」という問題に対

して既存の収集手法を発展させ、データ集約のみではなく、中継ノード上で映像を合成しつつ収集する手法を検討する。検討手法ではまず、撮影端末からのリアルタイム映像をクラウド上のノードで分担する。各映像の担当ノードは、自身の仮想リンクに基づいて、他の担当ノードへリアルタイム映像を転送し、同世界映像として配信を担当するノードまで最終的に転送する。この時、中継ノード上で複数の映像を合成し、合成後の映像のみを次のノードへ転送することで、通信負荷の削減や合成処理の分散を行える。

#### 4. LAN 環境での実装

本研究における前段階として、まず LAN 環境での同世界放送を実装した。これは、映像の収集と合成処理を分散させることによる効果を確認することを目的としたものである。図2に特定のノードのみが合成処理をする「集中合成」、図3に合成しつつ収集する「分散合成」を示す。図2および図3の4台の端末をそれぞれ A, B, C, D と区別する。いずれの端末も、各々のカメラで撮影した映像を用いる。図2の集中合成は、B, C, D から A に映像を集約させ合成する例である。図3の分散合成は、C の映像を A に転送すると同時に、D の映像を B に転送し、B で B と D の映像を合成し、その合成映像を A に転送し、A で最終的な合成映像を生成する例である。どちらの方法でも、A で映像の受信処理と合成処理をし、C, D では自身が撮影した映像を送信するのみである。しかし、集中合成では B が映像の送信のみを行うのに対し、分散合成では B が受信処理と合成処理もしている点が異なる。撮影は青の単色を背景としており、クロマキーによってあらかじめ用意した1つの背景画像に全ての映像を合成している。ただし、Bでの合成処理は、クロマキーによる合成を行っているのではなく、単に B 自身の映像と D から受信した映像を、サイズ調整をして1つの映像にしているだけである。本実装での映像処理は、背景の除去と合成のみであり、各被写体はあらかじめ指定された位置に表示されるため、ユーザが好きな場所に移動させることはできない。カメラからの映像の取得や処理は、コンピュータ・ビジョン・ライブラリの OpenCV によって行われる。そのため、映像の重なり合いは考慮していない。映像サイズは、カメラからの取得時は 640x480、同世界映像は 960x540 となっている。また通信は、IP アドレス、ポート番号を指定し、ソケット通信による TCP 通信を行っており、A, B, C, D は有線 LAN 接続されている。

#### 5. 評価

##### 5.1 実験環境

本実験では、1 フレーム当たりの合成にかかる遅延時間を計測するために、図4および図5に示すように遅延時間計測用の端末を無線 LAN で接続した。実験に使用する端

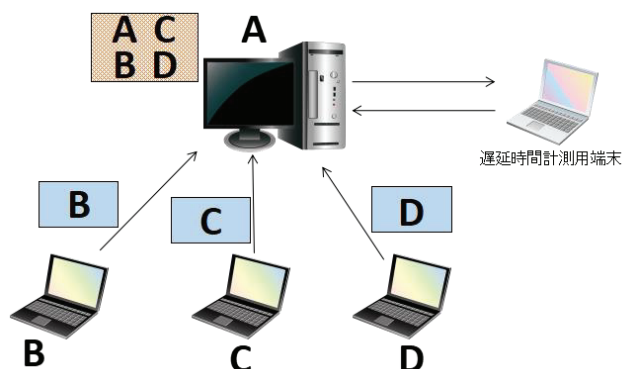


図4 集中合成の遅延時間計測方法

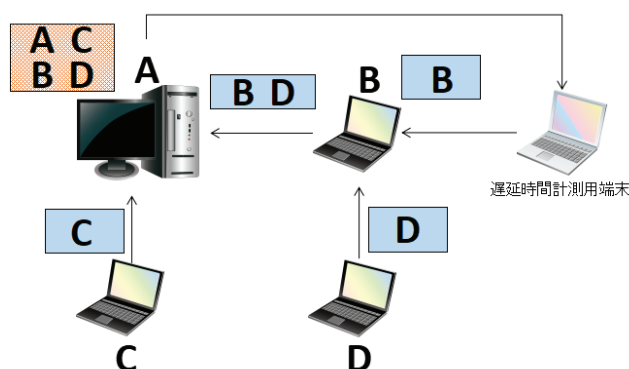


図5 分散合成の遅延時間計測方法

末として、図4および図5のAにWindows端末(OS: Windows10, CPU: AMD Ryzen 7 2700 8 core, 3.20GHz, メモリ: 32GB), BにWindows端末(OS: Windows10, CPU: Intel Core i5-7200U, 2 core, 2.50GHz, メモリ: 8GB), CおよびDにRaspberry Pi 3 Model B(OS: Raspbian, CPU: ARM Cortex-A53, 4 core, 1.2GHz, メモリ: 1GB), 遅延時間計測用端末にWindows端末(OS: Windows8.1, CPU: Intel Celeron CPU 847, 2 core, 1.1GHz, メモリ: 4GB)を使用した。

同世界放送では多数の接続を想定しているが、必要な端末数を用意することが困難であったため、本実験では1つの端末で複数のプロセスを実行し、疑似的に多接続を実現した。評価は、4接続から16接続まで、2接続間隔で、1フレーム当たりの各端末での処理時間と全体の遅延時間を計測した。4接続を超える場合、A, Bはカメラで撮影する1つの映像のみで、C, Dで複数プロセスを実行するようにした。2プロセス以上を実行する場合、実際にカメラで撮影する映像は1プロセスのみが使用し、他のプロセスは事前に撮影しておいた動画ファイルで代用した。映像の受信と合成を行う端末では、受信処理と合成処理が別プロセスで実行されており、受信処理のプロセス数は受信する映像の数に等しい。図6に、トランプを被写体とした16個の映像を合成した同世界映像のスクリーンショットを示す。この例では、画面を4x4の16分割として、各映像は決められた領域内に表示されている。1段目は左から順にA, B, C,



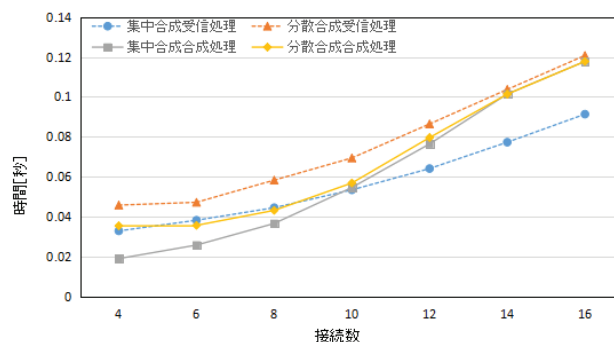
図6 同世界映像のスクリーンショット

D のカメラで撮影した映像，2 段目と 3 段目の左 2 つは C の動画ファイルの映像，3 段目の右 2 つと 4 段目は D の動画ファイルの映像が表示されている．動画ファイルはすべて同じものを使用したため，同じような映像が表示されている．

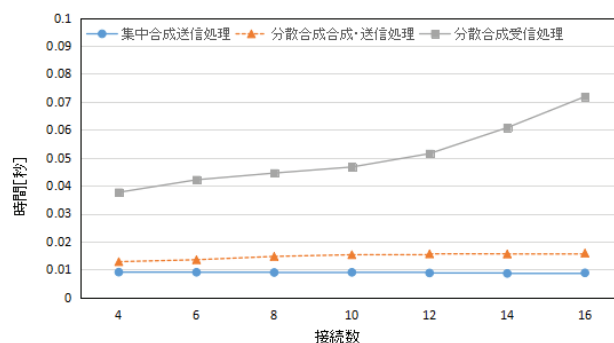
受信処理にかかる時間と合成処理にかかる時間は，それぞれで計測した．遅延時間計測用端末は，集中合成の場合も分散合成の場合も，0x01 の送受信によってタイミングを見るだけで，実際に映像を送り合うことはしなかった．また，端末の性能差による違いを確認するため，D に MacBook Pro (OS : OS X El Capitan, CPU : Intel Core i7, 4 core, 2.8GHz, メモリ : 16GB) を用いた場合も実験した．例えば，10 接続で実験する場合には，C および D の Raspberry Pi でそれぞれ 4 プロセスを実行する必要があるが，Raspberry Pi にかかる負荷が大きくなりボトルネックになり得るためである．D に MacBook Pro を用いる場合，Raspberry Pi にかかる負荷をさらに軽減するため，C を 1 台の Raspberry Pi で担当していたところを 2 台で分担させるようにした．また，MacBook Pro でカメラを用いて撮影する環境を作れなかったため，MacBook Pro ではすべての映像に動画ファイルを使用した．

## 5.2 実験結果

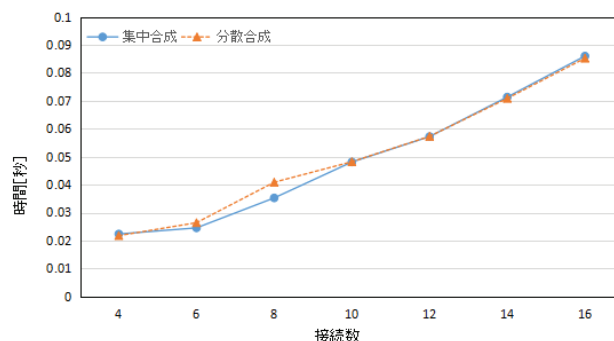
図 7 に MacBook Pro を用いなかった場合の結果を示す．集中合成および分散合成それぞれで各処理時間を 5 回ずつ測定し，その平均を示している．(a) A の処理時間は，各合成方法とも受信処理と合成処理にかかる時間を測定している．いずれも，接続数の増加に伴いほぼ一定の割合で処理時間が増加している．(b) B の処理時間は，集中合成では送信処理のみであるのに対し，分散合成では受信処理と合成・送信処理の 2 つの処理が存在する．合成と送信処理をまとめているのは，合成と送信が 1 つのプロセスで実行されているためである．B は接続数が増えても 1 つの映像を送ることしかしないので，集中合成送信処理は接続数に関わらず一定の処理時間となっている．また，分散合成合成・送信処理も接続数に関わらず一定となっているが，B での合成処理は自身の映像と受信した映像を 1 つの映像に



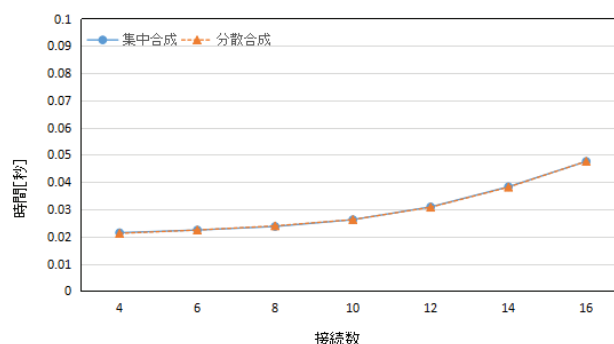
(a) A の処理時間



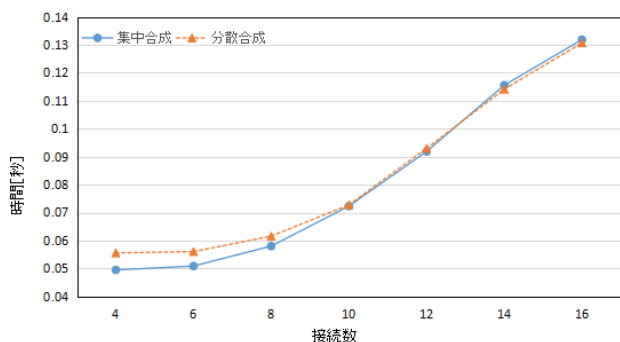
(b) B の処理時間



(c) C の処理時間



(d) D の処理時間

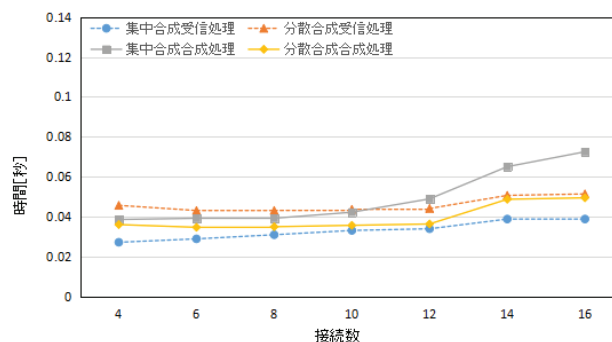


(c) 全体遅延時間

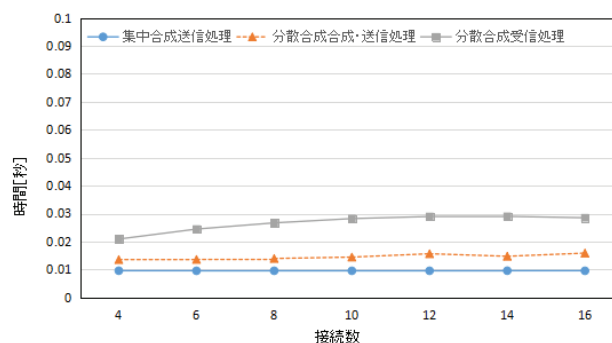
図7 MacBook Pro を用いない場合の実験結果

しているだけで、複雑な処理がなく負荷が小さいためと思われる。(c) C の処理時間は、接続数の増加に伴い処理時間が一定の割合で増加しているが、これは前述の通り、接続数を増やすために複数のプロセスを実行させたために負荷が大きくなることに起因していると思われる。集中合成と分散合成ではどちらも同じ動作をしているため、処理時間に差が見られない結果となっている。D も C と同じ動作をしているため、(d) D の処理時間も (c) と同様に集中合成と分散合成による処理時間の違いは見られない。(e) 全体遅延時間は、接続数が 4 の時と 6 の時はあまり変化がないが、それ以降は接続数の増加に伴い遅延時間も増加している。また、集中合成と分散合成による違いが見られない。つまり、映像の収集と合成を、1 台の端末ですべて行なった場合も 2 台の端末に分散させた場合も、処理にかかる時間が変わらないということである。この結果からは、本実験で期待した収集と合成方法の違いによる処理遅延の違いが確認できなかった。しかし、これは C, D に性能の低い Raspberry Pi を使用し、更に多数のプロセスを同時に実行したことによる処理の負荷が大きかったために、その影響が全体に及んでいるものと考えられる。

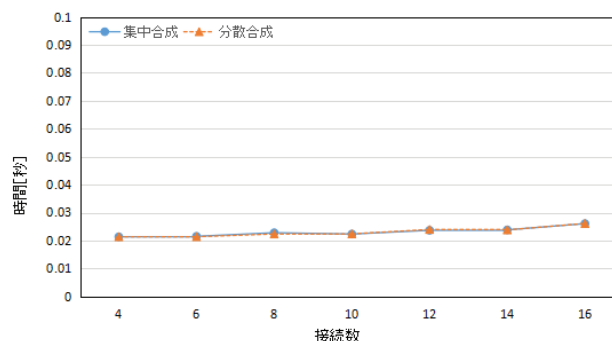
図 8 に MacBook Pro を用いた場合の結果を示す。この場合も、集中合成および分散合成それぞれで各処理時間を 5 回ずつ測定し、その平均を示している。A, B の処理内容に変化はないが、D に MacBook Pro を使用し、C を 2 台の Raspberry Pi で分担しており、それぞれ C1, C2 と区別している。そのため、C の処理時間も (c1) と (c2) に区別している。(a) A の処理時間は、集中合成合成処理について 10 接続以降で増加が見られ、その他では 12 接続以降で増加がみられる。(b) B の処理時間は、分散合成受信処理について、10 接続まで緩やかに増加し、それ以降は変化が見られない。また、図 7 の場合と同様に、集中合成送信処理と分散合成合成・送信処理は接続数に関わらず一定の時間となっている。(c1) および (c2) C の処理時間について、(c2) の 4 接続に記録がないが、これは C を 2 台で分担しているために起こっている。例えば、4 接続の時は C から



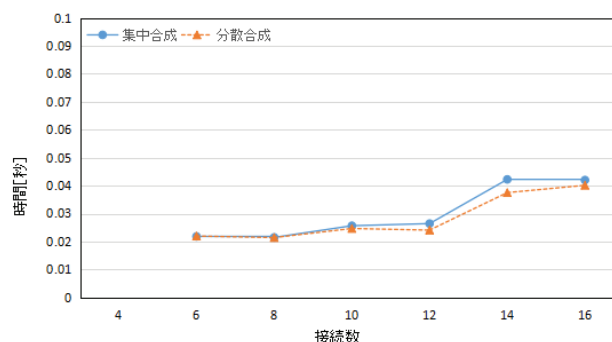
(a) A の処理時間



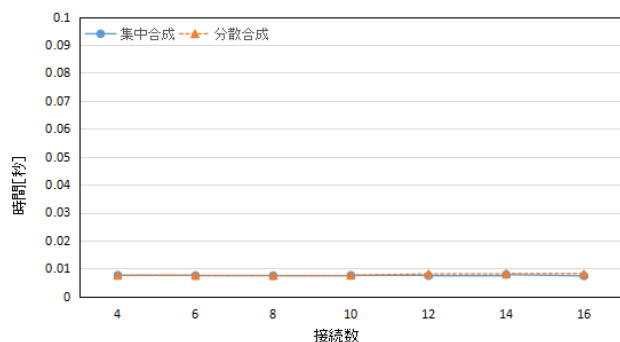
(b) B の処理時間



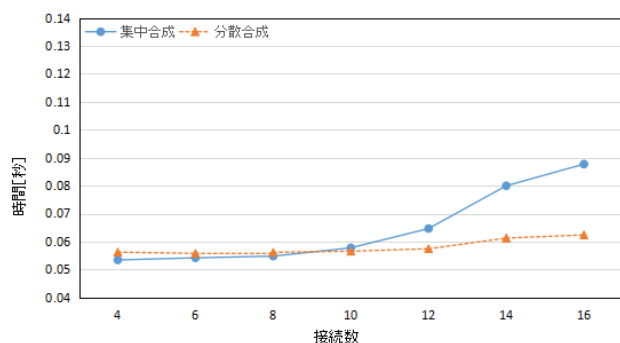
(c1) C1 の処理時間



(c2) C2 の処理時間



(d) D の処理時間



(e) 全体遅延時間

図8 MacBook Pro を用いた場合の実験結果

1つの映像を送信することになり、この時C1から映像を送信すれば、C2からは映像を送信する必要がない。6接続になると、Cから2つの映像を送信する必要があるため、C1とC2から1映像ずつ送信することになる。そのため、C1とC2各々は、接続数が4増える度に送信する映像が1つ増えることになる。(c2)の12接続から14接続で大きく処理時間が増加しているのもその為である。(d)Dの処理時間は、いずれの方法も変化がなく0.01秒未満で処理を終えている。(e)全体遅延時間は、集中合成は10接続以降に大きく増加しており、分散合成は12接続以降に若干増加しているがほぼ一定であるといえる。12接続以降の増加が目立つのは、(c2)で14接続になった時に処理時間が大きく増加していることが原因であると考えられる。図7の場合と異なり、接続数が増加すると分散合成の方が全体遅延時間を小さくできていることが確認でき、映像の収集と合成方法の違いによる差が見られた。

以上の結果より、送信端末での処理時間が大きいと、映像の収集・合成方法に関わらず全体の遅延が大きくなり、送信端末での処理時間が小さいなら、映像を合成しつつ収集の方が接続数が増えた時に全体の遅延が小さくなることが分かった。実際の使用環境では、1つの端末から1つの映像を送信することを想定するため、映像を合成しつつ収集することによって遅延時間を小さくする効果があると



図9 デモシステムのスクリーンショット

考えられる。

## 6. デモシステム

4章および5章のLAN環境での実装・評価に用いたシステムとは別に、デモ用のシステムを製作した。図9は、製作したデモシステムのスクリーンショットである。

4章の実装では、複数の映像を1つの背景に合成していたのだが、表示位置は固定されていた。デモシステムでは、本来想定される同世界放送により近づけるため、各映像の撮影ユーザが撮影対象を任意の位置に移動させて表示することができるようにした。各ユーザが表示位置を操作することになると、異なる映像が同じ位置に指定されることもある。その場合、どの映像が前面に表示されるのかが問題となるが、その表示順も各ユーザが奥行き情報として操作できるようにした。さらに、映像の拡大・縮小や回転も可能とである。拡大・縮小、回転、奥行きの3つの情報は、図9の右側にある赤・青・緑のパネルで操作できる。拡大・縮小に関しては、0.01~3.00までの倍率で調整する。回転に関しては、任意の整数値を入力し、その角度だけ回転させる。奥行きに関しては、0~100までの整数値を指定し、数値の大きい方が前面に表示される。これらの入力された数値は、全て合成サーバに送信され、その数値に基づいて同世界映像が生成される仕組みとなっている。

## 7. まとめ

本研究では、ライブ放送において全ての撮影対象がまるで同じ空間に存在するかのような同世界放送を提案し、そのシステムの検討とLAN環境での実装を行なった。複数のリアルタイム映像を1つの映像に合成する際に、特定のノードに処理が集中すると負荷が大きくなり、遅延が大きくなるため、中継ノードで映像を合成しつつ収集する必要がある。LAN環境での評価の結果、映像の合成処理を1つのノードに集中させる場合に比べて、映像を合成しつつ収集の方が全体の遅延時間を小さくすることができることが確認できた。

本実験は小規模なものであり表示位置も固定していたが、実際にはより多くの映像を用いた表示位置の移動に伴う重なり順も考慮する必要があるため、分散合成における映像

の収集方法に工夫が必要である。今後は、より実際の使用環境に近い環境で評価をするため、テストベッドを使用した実験を予定している。

## 謝辞

本研究の一部は、G-7 奨学財団研究開発助成事業および福井大学研究育成経費、JSPS 科研費 18K11316 の助成による成果である。

## 参考文献

- 1) NHK：中止の”すずめ踊り”合成動画作りました，知っトク東北，入手先 (<https://www.nhk.or.jp/sendai-blog/social/429730.html>) (参照 2020-07-31)。
- 2) 柿沼弘真ほか：機械学習を用いた任意背景リアルタイム被写体抽出技術，NTT 技術ジャーナル，Vol. 30, No. 10, pp. 16-20 (2018)
- 3) 高田英明：イマーシブテレプレゼンス技術”Kirari!”，日本画像学会誌，第 56 巻，第 4 号，pp. 366-373 (2017).
- 4) 阿久津明人，南憲一，日高浩太：超高臨場感通信技術 Kirari! Beyond 2020, NTT 技術ジャーナル，Vol. 30, No. 10, pp. 12-15 (2018).
- 5) 山口徹也ほか：高臨場感ライブビューイングサービスのためのサラウンド映像合成・同期伝送システム，映像情報メディア学会誌，Vol. 74, No. 2, pp. 402-411 (2020).
- 6) 佐藤孝子ほか：競技空間全体の高臨場ライブ中継に向けたサラウンド映像合成・同期伝送技術，NTT 技術ジャーナル，Vol. 29, No. 10, pp. 19-23 (2017).
- 7) 野中敬介，渡邊良亮，塚本航平：5G 網を利用した自由視点映像リアルタイム配信技術，映像情報メディア学会誌，Vol. 74, No. 1, pp. 180-186 (2020).
- 8) 加藤大一郎，武藤一利，三ッ峰秀樹：ハイブリッドセンサを用いたハンディカメラによるバーチャルスタジオの開発と実用化，映像情報メディア学会誌，Vol. 72, No. 1, pp. J13-J22 (2018).
- 9) M. Keshavarzi et al.: Optimization and Manipulation of Contextual Mutual Spaces for Multi-User Virtual and Augmented Reality Interaction, Proc. of the 2020 IEEE VR 2020, 10 pages (Mar. 2020).
- 10) 田中一品，加藤良治，中西秀之：鏡型ビデオ会議における視覚相互作用によるソーシャルテレプレゼンスの強化，情報処理学会論文誌，Vol. 58, No. 5, pp. 946-954 (May 2017).
- 11) T. Pejsa et al.: Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment, Proceedings of the 19<sup>th</sup> ACM Conference on Computer-Supported Cooperative Work & Social Computing, pp. 1716-1725 (2016).
- 12) Jaron Lanier: How to get the most from Together mode, Microsoft, available from (<https://techcommunity.microsoft.com/t5/microsoft-teams-blog/how-to-get-the-most-from-together-mode/ba-p/1509496>) (accessed 2020-08-19)
- 13) Apple Inc., Jean-Francois M Albouze, Santa Cruz: Generating synthetic group selfies, US pat. 10,672,167, 2020
- 14) S. Matsumoto, T. Yoshihisa, T. Kawakami, Y. Ishi, Y. Teranishi: Different Worlds Broadcasting: A Distributed Internet Live Broadcasting System with Video and Audio Effects, Proceedings of the 31st IEEE International Conference on Advanced Information Networking and Applications (AINA 2017), pp. 71-78 (2017).
- 15) 義久智樹，川上朋也，石芳正，寺西祐一：異世界放送：映像効果と音声効果を伴う分散型インターネットライブ放送システム，マルチメディア，分散，協調とモバイル (DICOMO2016) シンポジウム，pp. 1827-1832 (2016).
- 16) T. Kawakami, S. Matsumoto, Y. Ishi, T. Yoshihisa, Y. Teranishi: An Implementation of a Rule-Based Distributed Video Processing System, Proceedings of the 23rd IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN 2017) Demos, 2 pages (2017).