

# 背景置換処理のオフローディングによる ARシステムの実装評価

坪木 良宏<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>

**概要:** 近年の VR (Virtual Reality) や AR (Augmented Reality) の技術の進歩により, ユーザーは高品質な仮想世界を体験することができるようになった. VR アプリケーションでは, 現実世界の周囲の状況が見えないため, 一般にユーザーの身体的な動きが制限されます. AR は物理的な動きの制限されずに仮想世界を体験することができるが, 仮想世界として置き換えられる範囲は限定的である. 本研究では, スマートフォンやタブレット端末などのモバイル端末の利用を想定し, カメラで撮影したリアルタイムの実世界映像から背景部分のみを除去し, 除去した背景部分のみを仮想的な背景に置き換える, 部分的な仮想世界システムを実装し, 評価を行った.

## 1. はじめに

近年の VR (Virtual Reality) や AR (Augmented Reality) の技術の進歩により, 教育 [1, 2], ショッピング [3, 4], 観光 [5, 6] などの様々な分野で仮想世界を体験できるようになってきている. それらの VR/AR サービスの中でも我々は, ユーザーが別の時間や空間の中にいるかのような仮想世界体験 (時空間移動体験と呼ぶ) を提供するシステムに着目した. これらのような VR/AR サービスでは, 現実世界でのユーザーの物理的な行動を制限されないことが望ましい. つまりユーザーが仮想世界を体験しながら, 現実空間内を自由に歩き回れるシステムであることが望まれる.

VR では, ユーザーはヘッドマウントディスプレイ (HMD) を装着し, ディスプレイを見ることで仮想世界を体験することができる. HMD に搭載されているセンサーがユーザーの頭の向きや動きを検知することで, ディスプレイに表示される仮想世界の視点の角度や向きも同様に变化する. しかしながら, ディスプレイに表示される映像はユーザーのいる現実世界の景色とは完全に異なるため, ユーザーは移動することにより現実世界の周囲に存在するモノや人にぶつかってしまう恐れや, 転倒してしまう恐れがある. そのため, 一般的にユーザーの移動は制限される

一方で, 既存の AR では仮想オブジェクトが現実世界の上に重なるようにして表示される. そのため, ユーザーは仮想オブジェクトと同時に現実世界の様子も確認することが可能なので, 自由に動き回ることができる. しかしながら, AR ではユーザーが異なる時空間にいるかのような体験をすることはできないため, ユーザーの仮想世界体験は限定的なものになってしまう.

本研究では, ユーザーのいる屋内の空間に異なる時空間の仮想世界を重ね合わせることで, 時空間移動体験を提供するシステムを目指す. 特別な部屋を用いて VR 映像を投影するシステムがある [7]. このシステムを使用することで, 部屋全体がまるで別の時空間に移動したかのような体験が可能になります. しかしながら, このようなシステムでは VR 映像を部屋全体に投影するような特別な設備や施設が必要となるため, コストがかかります. そのような特別な設備や施設を用意することなく, HMD やモバイル端末を用いて, AR アプリケーションとして時空間移動体験を提供することを目的とする.

提案システムによる時空間移動体験イメージを図 1 に示す. ユーザーが HMD やモバイル端末のディスプレイをとおして現実世界の周囲を見渡すと, 屋内空間が部分的に仮想世界にリアルタイムで置き換わる. ユーザーは周囲のモノや人にぶつかってしまう恐れなく仮想世界体験をすることが可能になる.

本稿では, 屋内空間の一部を背景と見なし, その背景を仮想世界にリアルタイムで置き換える AR アプリケーションを実現するシステムを提案する. 一般に, すべての HMD やモバイル端末が屋内空間の一部を背景と見なし, その背景

<sup>1</sup> 福井大学大学院工学研究科  
Graduate School of Engineering, University of Fukui

<sup>2</sup> 大阪大学サイバーメディアセンター  
Cybermedia Center, Osaka University

<sup>3</sup> 国立研究開発法人情報通信研究機構  
National Institute of Information and Communications  
Technology



図 1 提案システムによる時空間移動体験イメージ (左: 使用前, 右: 使用後)

をリアルタイムに仮想世界に置き換えるような処理をするのに十分な処理能力を持っているとは言えず、むしろ遅延の影響を受けやすい処理といえる。リアルタイムでスムーズな処理を行うために背景の置換処理をエッジ/クラウドサーバーにオフロードすることを想定している。

以下本稿では、2章で関連研究を述べ、3章で背景領域の決定、置換の手法について説明し、4章で提案手法の実装・評価について述べたのち、最後に5章でまとめを述べる。

## 2. 関連研究

リアルタイム拡張現実システムに関する類似研究がある。木戸らは深層学習によるセマンティックセグメンテーションを行い、現実のシーンにバーチャルコンテンツを重ね合わせる複合現実 (Mixed Reality, MR) ベースの未来景観を評価するシステムを開発することで建造物建設の際に景観評価に携わる関係者間のコンセンサスを得ることを可能にした [8]。このシステムではカメラから 100m 離れた場所にある物体と AR オブジェクトの動的オクルージョン処理を実現することにより、屋外の景観に建設予定の建造物を合成表示させている。この研究ではセマンティックセグメンテーションをもちいてセグメンテーションされた領域を前景や背景に分類し、AR オブジェクトと現実世界の景観とのリアルタイムなオクルージョン処理を行っているが、セマンティックセグメンテーションの処理が重く、フレームレートが約 35 という値になってしまうということからリアルタイムで実行するシステムとしては不十分という問題があった。また、現実世界の景観に AR オブジェクトを違和感なく合成するを目的とするこのシステムは、本研究の AR 室内旅行システムのユーザーの付近は現実世界のまま、ユーザーから一定距離以上の空間をすべて任意の背景に置換するという目的とは大きく異なる。

文献 [9] では、自然特徴点からなるランドマークデータベースを事前に構築し、データベースと入力画像中の自然特徴点を対応づけることによりカメラの位置、姿勢を推定

することで正しい位置にコンピュータグラフィック (CG) を重畳表示させる拡張現実感システムを実現させた。[9]の研究では、複雑な広域環境での推定誤差の少ない高精度なカメラ位置・姿勢推定手法による CG の重畳表示を実現し、ウェアラブル拡張現実システムへの応用の可能性を示したが、実時間処理の点に関しては保留されていたため、こちらの研究もリアルタイムで実行可能なシステムとまでは言えないという問題があった。また自然特徴点を取得するには事前に特別なセンサーなどを用いてシステムを使用する環境のデータベースを構築する必要があるため新しい環境でシステムを導入する場合、システムを使用可能な環境を整えるためにかなりの時間を要すると考えられる。

Atmosph Window [10] は、室内に設置する窓型のディスプレイ装置であり、ディスプレイに異なる空間を映し出すことで、異なる空間に存在する部屋に居る感覚を得られるようにする。ディスプレイ上の映像は、ユーザの顔位置を検出した上で角度に応じて変化し、視点変化に追従する。しかし、ユーザの体験はこの窓型の装置が置かれた付近のみになってしまうため、かなり限定的な仮想世界体験になってしまう。

バーチャル平安京 AR [11] は AR 技術を用いて、平安京の景観を現代によみがえらせるスマートフォンアプリケーションである。ユーザーはスマートフォンを現実世界にかざすことで、平安時代の景観が重畳表示され、平安時代にタイムスリップした可能様な体験が可能である。しかしながら、重畳表示する AR オブジェクトを事前に作成する必要があり、また AR によって現実世界の拡張体験ができるのは、平安京跡地のみに限られる。

新型コロナウイルス (COVID-19) の流行により近年多く使用されるようになっている Zoom, WebEx, Microsoft Teams などリモート会議システムには、仮想的な背景を設定する機能がある。これは、映像の中から人間の領域を検出し、それ以外の領域を背景としてみなすものである。このような機能は、一般に Deep Image Matting と呼ばれる物体の切り抜き処理によって実装されている。これまでにも、ディープラーニングを用いて高精度の物体の切り抜きを実現する研究はいくつか行われてきた [12]。しかし、それらの技術は、リモート会議中の人物の領域を検出して、背景を置き換えるような限られた状況にしか適用できない。

## 3. 提案システム

提案システムの概要図を 2 に示す。提案システムでは、はじめにユーザーがカメラの搭載されているデバイス (HMD やスマートフォン) で現実世界の映像を撮影する。撮影された映像をフレーム単位で背景置換を行い、背景置換後のフレームをユーザーのデバイスに送り返し表示する。AR では一般的に、Light Detection And Ranging (LiDAR) のような赤外線センサーや AR マーカーを用いて距離や空間情

報などを取得する。しかし、その場合 LiDAR のようなセンサーの搭載されていない端末では空間情報を認識することができず、AR グラスなどのウェアラブル端末にセンサーを搭載する場合、デバイスがより大きく重くなり、実用性の低下につながります。また AR マーカーの場合、マーカーを設置することのできない場所の空間情報の認識は不可能となる。

そこで、ユーザーの移動が制限されない時空間移動体験を実現するため、我々は機械学習による深度推定に基づいたリアルタイム背景置換手法を用いた。図 3 に背景置換処理の流れを示す。提案手法では、ユーザーがモバイル端末で撮影した映像をエッジ/クラウドサーバーで受け取り、背景として置き換えたい領域を検出する処理をフレーム単位で行う。背景領域の検出には 2 つの処理を用いている。1 つは機械学習による深度推定処理、もう 1 つはエッジ検出処理である。これら 2 つの処理結果をもとに、受信したフレームを背景領域と非背景領域の 2 つの領域に画素単位で分割している。

はじめに、機械学習による深度推定処理について説明する。入力画像に対し、機械学習による深度推定をおこなうと図 4 に示すような深度推定画像を得る。この深度推定画像の各画素は深度を表す数値を保持しており、深度が深いほど黒く、浅いほど白く表示される。つまり深度値が大きいほど黒く、深度値が低いほど白くなるような深度推定画像が生成される。画素の深度値をあらかじめ決定した閾値と比較し、深度値が閾値より大きい場合、その画素を背景領域であると判定する。モバイル端末や AR デバイスには、LiDAR などの深度センサーが搭載されているものもあるが、本研究では、カメラを搭載されているモバイル端末であれば、汎用的に使用できるシステムとするため、機械学習による深度推定を用いて空間情報の認識を行った。

次に、エッジ検出処理について説明する。入力画像に Canny 法と呼ばれるエッジ検出処理を行うと図 5 のようなエッジ検出画像を得る。得られたエッジ検出画像にモルフロジー処理と呼ばれる画像の収縮処理と膨張処理を複数回ずつ行うことで、エッジ検出画像内のノイズを除去した画像を得る(図 6)。深度推定画像の画素の保持する深度値が閾値付近であるような画素についてはエッジ検出処理によって得られた結果(図 6)を用いて背景領域/非背景領域の判定をすることで、高い精度で分割を行う。

これら 2 つの処理結果により、図 7 に示すような背景領域と非背景領域に分割されたマスク画像を生成する。そして、生成されたマスク画像を用いて、背景をユーザーのデバイスの角度や向きに応じた仮想世界へと置き換える。

#### 4. 実装・評価

前章で述べた提案手法を用いた AR アプリケーションを実装し、実装したアプリケーションを用いた実験による評

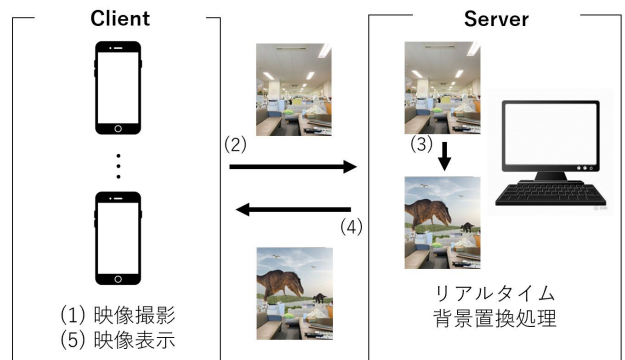


図 2 システム構成

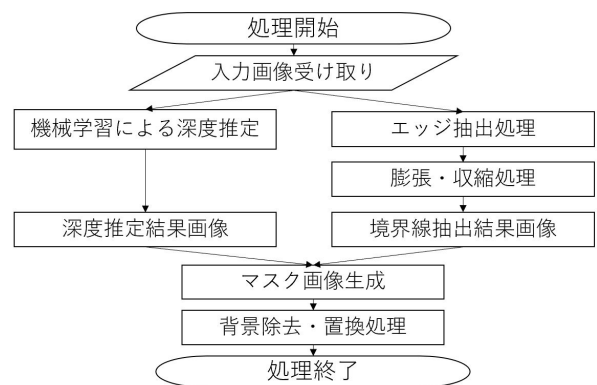


図 3 背景置換処理のフローチャート



図 4 深度推定画像

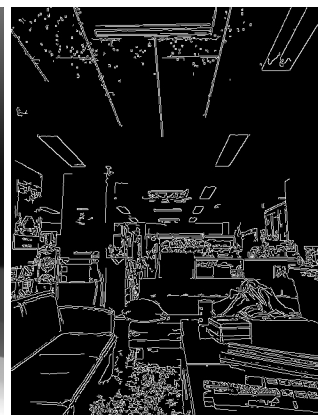


図 5 エッジ検出画像

価を行った。評価では、背景領域の判定精度の測定と応答時間の計測を行った。

システム構成は図 2 で示したとおりで、映像を撮影するクライアントにはモバイル端末を、機械学習による推定深度に基づいた得られた画像に対する背景置換処理は処理性能の高い 1 台のデスクトップ PC にオフロードしている。画像処理をサーバー側のデスクトップ PC にオフローディングすることにより、クライアントとして使用するモバイル端末の処理能力が低い場合であっても、リアルタイムでの画像処理を可能にしている。

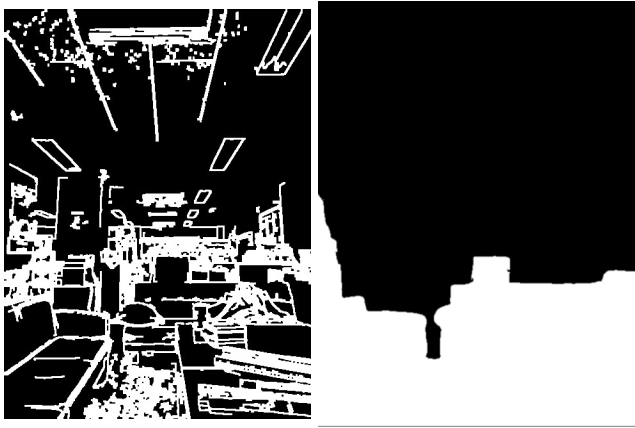


図 6 モルフォロジー処理画像

図 7 マスク画像

#### 4.1 実験環境

表 1 機器詳細

機器	詳細
デスクトップ PC(Server)	OS: Ubuntu20.04, CPU: Intel Core i5-10210U, 8 core, 3.1GHz, Memory: 32GB, GPU: GeForce RTX3060 Ti
Raspberry Pi 4(Client)	OS: Raspbian10, CPU: ARM Cortex-A72, 4 core, 1.5GHz, Memory: 4GB, GPU: VideoCore VI

実験は1台のデスクトップPCとRaspberry Pi 4 Model Bを4台用いて行った。クライアント端末として使用するRaspberry Pi 4では現実世界の映像を撮影し、サーバーに送信するため、カメラモジュールを搭載している。またサーバー側で用いるデスクトップPCには高速な画像処理を可能とするため、外付けGPUとしてGeForce RTX3060 Tiを用いている。クライアント側とサーバー側のソフトウェアはPythonによって実装されており、Raspberry Pi 4とサーバーのデスクトップPC間はIEEE 802.11acの無線LANで接続されている。また、クライアント-サーバー間の映像の通信にはFFmpegと呼ばれるソフトウェアを使用した。

今回の実験では、同時に使用するクライアント端末数と、画像サイズによる変化を評価するためRaspberry Pi 4を、1-4台使用し、それぞれ480×360, 240×180, and 120×90の3パターンの画像サイズで実験を行った。

機械学習による深度推定処理に用いる、学習済みモデルにはRenéらの学習済み深度推定モデルであるMiDaSを用いた[13]。学習済みの深度推定モデルはいくつも公開されているが、今回MiDaSを用いた理由として、提案システムではリアルタイム背景除去を行う必要があったため、高速な深度推定が可能である学習済み深度推定モデルMiDaS

を用いた。MiDaSにはMiDaS v2.1 - Small, MiDaS v2.1 - Hybrid, MiDaS v2.1 - Largeの三つのモデルが用意されており、そのうちの一つであるMiDaS v2.1 - Smallは推定精度こそ他の二つのモデルより低精度ではあるが、高速でリアルタイムシステム向けのモデルとなっている。そのためリアルタイムアプリケーションを想定している今回の実装では、そのMiDaS v2.1 - Smallというモデルを用いた。また、学習済み深度推定モデルであるMiDaSでは、機械学習の際のトレーニングデータセットに通常の深度画像データセットだけでなく、3D映画によって作成されるデータセットが採用されており、街の景観から自然の景観まで多種多様な景観、動いている物体に対する深度推定が他の機械学習による深度推定モデルより高い精度であったことから、本システムのような動的シーンでの安定した深度推定が可能であるということもありMiDaSを用いた。

エッジ検出処理には、コンピュータ・ビジョン・ライブラリであるOpenCV[14]のcv2.Canny()関数を用いた。エッジ検出処理にはv2.dilate(), cv2.erode()関数を使用した。エッジ検出処理にはCanny法のほかにも、SobelフィルタやLaplacianフィルタを用いたエッジ抽出が存在するが、今回用いたCanny法によるエッジ抽出が最もノイズが少なくエッジを抽出することが可能であったことから、Canny法によるエッジ抽出を用いた。なお、映像の読み込みもOpenCVによって行っている。

#### 4.2 背景置換処理の精度

背景除去の精度の評価指標については、画像処理の分野で精度の評価の際に用いられることの多い、式(1),(2)によって計算される再現率(Recall)と適合率(Precision)を用いた。

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

ここで、TP(true positive)は実際背景で背景と判断された画素数、FP(false positive)は実際は非背景で背景と判断された画素数、FN(false negative)は実際背景で非背景と判断された画素数、TN(true negative)は実際非背景で非背景と判断された画素数を表している。

評価指標を用いた背景置換精度の具体的な評価方法を説明する。実装したシステムによって得られる背景除去後のフレームを無作為に三つ抽出する。その抽出した三つのフレームに対する背景領域の正解画像を手動で作成する。抽出した三つのフレームをそれぞれの正解画像と比較し、再現率と適合率を計算し、計算された再現率の平均と適合率の平均の結果から評価を行った。背景領域の正解画像については、深度センサーを用いて作成することも考えられた

が、今回はユーザー視点で自然な背景領域を評価したいため、人の手による手動の正解画像作成を行った。

本章の冒頭で述べた実験環境を用いて、480×360、240×180、120×90の各画像サイズでの背景置換精度を評価するための提案手法によって得られたマスク画像と背景領域の正解画像の比較を図8に、評価指標の数値結果を表2に示す。図8では、最上の上から順に、マスク画像を生成する際に用いた入力フレーム、入力フレームから手動で作成した背景領域の正解画像、提案手法により入力フレームから生成されたマスク画像となっており、左から順に、画像サイズ480×360、240×180、120×90の結果となっている。

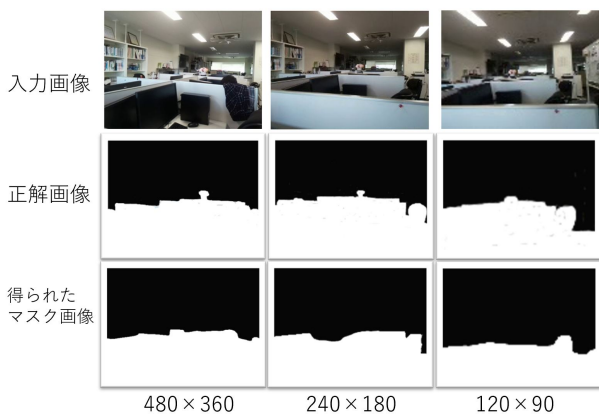


図8 各画像サイズでのマスク画像比較

表2 各画像サイズでの背景置換精度比較

	Image size		
	480×360	240×180	120×90
Recall	1.00	1.00	1.00
Precision	0.950	0.890	0.869

表2の結果から、3つすべての画像サイズで再現率は1.00であったことから、今回実験したすべての画像サイズで実際に背景である領域を取りこぼしなく背景領域として判断できていることが確認できる。また、画像サイズによる再現率の差は無かったことから実際に背景である領域を非背景と判断してしまう取りこぼしは、画像サイズの変化によっては影響は受けないと考えることができる。また適合率の結果を見ると画像サイズが小さくなるにつれてわずかに適合率は下がっているものの、すべての画像サイズで0.8以上の高い値であったことから、非背景である領域を背景領域と判断してしまう背景の誤検出が少ないことを確認できた。画像サイズが小さくなるにつれ、適合率が下がっていることについては、画像サイズを小さくすると機械学習による深度推定の精度が落ちるため、機械学習による深度推定結果に依存するマスク画像で背景の誤検出が増えてしまい、適合率が低下したと考えることができる。

2つの指標を用いた評価結果から提案手法によるマスク画像を用いた背景除去の精度は十分であることが分かった。また、入力映像の画像サイズを下げすぎると機械学習による深度推定の精度が落ちることにより、背景の誤検出が増えると考えられ、適合率が低下することが分かった。

### 4.3 リアルタイム性の評価

システムのリアルタイム可用性の評価する際の指標には、1フレームあたりの処理時間と映像遅延を用いた。ここで1フレームあたりの処理時間というのは、モバイル端末で撮影された映像がサーバー側で処理される際に、1フレームの入力画像に機械学習による深度推定と画像処理を用いて、背景を除去、置換にかかった処理時間のことを指している。また、ここでの映像遅延というのはクライアント側で撮影された映像のなかの任意のフレームが、サーバー側に渡され処理された後、再びサーバー側からクライアント側に送り返され表示されるまでに要する時間のことを意味している。

1フレームあたりの処理時間が大きい場合、クライアント側で表示される映像のフレームレートが低くなり、ユーザー視点の背景置換映像が映像のかくつきなどから自然な映像とは言えなくなってしまうため、ユーザー視点の背景置換映像が違和感を感じない滑らかな映像となるように1フレームあたりの処理時間は短くする必要がある。また、映像遅延が大きい場合は、ユーザーの視線方向が変化した際に、モバイル端末に表示されている映像が正しい視線方向の映像へ変化するのが遅れ、ユーザーが違和感を感じることに繋がってしまうため、なるべく映像遅延は小さく抑えることがリアルタイムアプリケーションとして求められる。

図9は、クライアントとして使用するRaspberry Piの台数と撮影する動画の画像サイズをそれぞれ変化させたときの1フレームあたりの処理時間の結果を示しています。

比較のため、Raspberry Pi単体で処理を行った場合の1フレームあたりの処理時間を表に示す。表3と図9の結果を比較すると、3つの画像サイズとも1フレームあたりの処理時間が大幅に短縮されていることが分かります。また、いずれの場合も1フレームあたりの処理時間は140[ms]以下であり、リアルタイムで背景の置き換えが出来ていることがわかる。

当然ことながら、画像サイズが大きくなればなるほど、1フレームあたりの処理画素数は多くなるため、本実験で最も大きい画像サイズである480×360の場合は、他の2つの画像サイズに比べて1フレームあたりの処理時間が大きくなっている。一方、本実験で最も小さい画像サイズである120×90の場合、1フレームあたりの処理時間は小さくなっている。また、横軸に設定した使用クライアント端末の台数による1フレームあたりの処理時間の変化を見ると、どの画像サイズの場合でも、撮影端末の台数を増やすと1

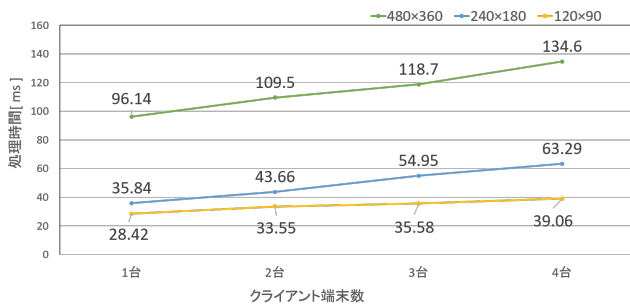


図 9 1 フレームあたりの処理時間結果

フレームあたりの処理時間が長くなることが確認できる。

表 3 Raspberry Pi 単体で実装した場合の処理時間

指標	画像サイズ		
	480×360	240×180	120×90
処理時間 [ ms ]	869.5	434.8	306.7

図 10 は撮影に使用した Raspberry Pi の台数と撮影する映像の画像サイズをそれぞれ変更した際の映像遅延の結果を示している。

1 フレームあたりの処理時間と同様に、比較のために、Raspberry Pi 単体で処理した場合の映像遅延の結果を表 4 に示します。表 4 と図 ??の結果を比較すると本システムを使用することですべての画像サイズで映像遅延が半分以下となっていることが確認できます。

1 フレームあたりの処理時間とレイテンシーの 2 つの評価結果から、本システムは推定深度に基づくリアルタイム背景置換を十分に行えていると言える。

表 4 Raspberry Pi 単体で実装した場合の映像遅延

指標	画像サイズ		
	480×360	240×180	120×90
映像遅延 [ ms ]	867.4	435.2	308.1

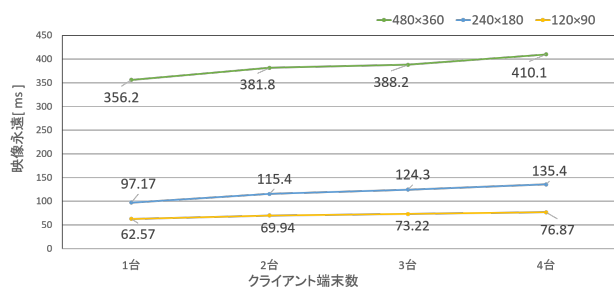


図 10 映像遅延結果

## 5. まとめ

本研究では、屋内空間の一部を背景とし、その背景をリアルタイムに仮想空間の景観に置き換える時空間移動体験のための AR アプリケーションを実現するシステムを提案し、実装・評価を行った。本システムでは、機械学習による深度推定や背景置換などのすべての画像処理をエッジ/クラウドサーバーにオフロードすることで、リアルタイムでの時空間移動体験を実現した。

提案システムの背景置換精度とリアルタイムアプリケーションとしての可用性を評価するため、カメラの搭載されたモバイル端末 1-4 台使用し、3 パターンの画像サイズで実験を行った。背景置換精度の評価結果から、十分な精度で背景置換が行えていることを確認した。また、リアルタイムアプリケーションとしての可用性の評価から、複数のクライアントを同時に使用した場合でも、リアルタイムで背景の置き換えが可能であることを確認した。

しかしながら、まだいくつかの課題も残されている。まずはじめに、背景領域の判定に用いている深度閾値を自然に決定することが挙げられる。現在のシステムでは深度閾値はあらかじめ決定されたものを使用しており、静的な値である。しかし、深度閾値はシステムを使用する部屋の形状に応じて、動的に調整されることがより効果的であると考えられる。また、処理時間と画像サイズのトレードオフが課題として挙げられる。評価結果からは、画像サイズが大きくなるほど、処理時間や映像遅延は大きくなることが分かった。よりよい仮想世界体験実現するために、高画質の場合でも、処理時間や映像遅延を小さく抑えることを可能にするため、性能向上が求められる。また、現在のシステムでは置換される仮想世界の景観はユーザーの頭の向きが変化した場合でも移り変わらないため、VR のようにユーザーの動きに合わせて、自然に移り変わる仮想世界を表示させることも今後の課題である。

## 謝辞

本研究の一部は JSPS 科研費 22K12009, 21H03429, 18K11316, 放送文化基金, 「北陸地域の活性化」に関する研究助成事業, G-7 奨学財団研究開発助成事業による成果である。

## 参考文献

- [1] Scavarelli, A., Arya, A. and Teather, R. J.: Virtual reality and augmented reality in social learning spaces: a literature review, *Virtual Reality*, Vol. 25, No. 1, pp. 257–277 (2021).
- [2] Kamińska, D., Sapiński, T., Wiak, S., Tik, T., Haamer, R. E., Avots, E., Helmi, A., Ozcinar, C. and Anbarjafari, G.: Virtual reality and its applications in education: Survey, *Information*, Vol. 10, No. 10, p. 318 (2019).
- [3] Xi, N. and Hamari, J.: Shopping in virtual reality: A lit-

- erature review and future agenda, *Journal of Business Research*, Vol. 134, pp. 37–58 (2021).
- [4] Smink, A. R., van Reijmersdal, E. A., van Noort, G. and Neijens, P. C.: Shopping in augmented reality: The effects of spatial presence, personalization and intrusiveness on app and brand responses, *Journal of Business Research*, Vol. 118, pp. 474–485 (2020).
- [5] Beck, J., Rainoldi, M. and Egger, R.: Virtual reality in tourism: a state-of-the-art review, *Tourism Review* (2019).
- [6] Azevedo, J. N. and Alturas, B.: The augmented reality in Lisbon tourism, *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE, pp. 1–4 (2019).
- [7] Manjrekar, S., Sandilya, S., Bhosale, D., Kanchi, S., Pitkar, A. and Gondhalekar, M.: CAVE: an emerging immersive technology—a review, *2014 uksim-amss 16th international conference on computer modelling and simulation*, IEEE, pp. 131–136 (2014).
- [8] Kido, D., Fukuda, T. and Yabuki, N.: Assessing Future Landscapes Using Enhanced Mixed Reality with Semantic Segmentation by Deep Learning, *Advanced Engineering Informatics*, Vol. 148, No. 101281, pp. 1–13 (2021).
- [9] 大江統子, 佐藤智和, 横矢直和: 幾何学的位置合わせのための自然特徴点ランドマークデータベースを用いたカメラ位置・姿勢推定, Vol. 10, No. 3, pp. 285–294 (2005).
- [10] 姜 京日: 本当の夢を探し続けよう, *感性工学*, Vol. 19, No. 2, p. 87 (2021).
- [11] 矢野佳司: バーチャル平安京 AR, note = , <http://www.ritsumeai.ac.jp/news/detail/?id=1067> , OPTmemo = .
- [12] Xu, N., Price, B., Cohen, S. and Huang, T.: Deep Image Matting, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2970–2979 (2017).
- [13] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K. and Koltun, V.: Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14 (2020).
- [14] : OpenCV. , <https://opencv.org/>.