

実・仮想空間を跨るインタラクションを実現するための QoS 適応機構とその評価

山本 眞也^{†1} 村田 佳洋^{†2} 柴田 直樹^{†3}
安本 慶一^{†1} 伊藤 実^{†1}

実空間のモバイルユーザと遠隔のネットワーク・ユーザの間のスムーズな共同作業およびインタラクションを実現するには、ユーザがリアルタイムに動くオブジェクトを観測できるように、各オブジェクトの位置と方向の情報（AR 情報と呼ぶ）をリアルタイムに配送する必要がある。しかし、多数の動くオブジェクトを対象にする場合、無線 LAN などの帯域幅の限られた環境下で AR 情報をリアルタイムに配送するのは、しばしば困難である。そこで、本論文では、ユーザごとに、ユーザにとってより重要なオブジェクトがより高い頻度で更新されるよう AR 情報の配送頻度を自動調整する QoS 適応機構を提案する。評価実験の結果、提案機構により、一般的なネットワーク環境を用いたモバイルユーザと遠隔ユーザの環境上で実用的な更新レートおよび短い遅延時間でオブジェクトが更新されることを確認した。

QoS Adaptation Mechanism to Realize Interaction between Virtual and Real Worlds and its Evaluation

SHINYA YAMAMOTO,^{†1} YOSHIHIRO MURATA,^{†2}
NAOKI SHIBATA,^{†3} KEIICHI YASUMOTO^{†1}
and MINORU ITO^{†1}

In order to realize smooth cooperative work and interaction between mobile users in the real world and remote network users, it is required to allow mobile users and remote users to share the same view of a shared space including many moving objects in sufficient quality for interaction. For this purpose, information (called AR information) of the position and orientation of each object must be delivered to user terminals, so that the users can see moving objects updated in real-time. However, it is difficult to deliver AR information in real-time due to limited network bandwidth in an ordinary wireless LAN or Internet environment. In this paper, we propose a QoS adaptation mechanism which allows users to see more important objects with a higher update frequency. Through

experiments on network environments with various available bandwidths, we confirmed that the proposed mechanism achieves a practical framerate of object update between mobile users and remote users with reasonably short delay in pervasive network environment.

1. はじめに

近年、様々な物体（以下、オブジェクトと呼ぶ）をユーザの目の前にあたかも存在するように表示する MR (Mixed Reality) 技術や AR (Augmented Reality) 技術が注目されている¹⁾⁻³⁾。また、遠距離にいる複数のユーザがネットワークを介して参加し協調作業を行う NVE (Networked Virtual Environment) や CSCW (Computer Supported Cooperative Work) に関する研究もさかんに行われている^{4),5)}。これらの技術をあわせて用いることで、現実空間で行われている様々な社会活動（たとえば、コンサート、ショッピング、展示会、スポーツ、ゲーム、旅行、講習会、会議、共同作業など）に、遠隔ユーザ（以下、仮想ユーザと呼ぶ）がネットワークを介して仮想的に参加することが可能になる。このような実・仮想ユーザ混合対話型アプリケーションにおいて、実ユーザおよび仮想ユーザが円滑かつリアルに協調作業できるようにするには、(1) 現実空間から参加しているユーザ（以下、実ユーザと呼ぶ）と仮想ユーザが同じ空間を共有できること、(2) 各ユーザが共有空間内を自由に移動でき、それぞれの空間内の位置や向いている方向に応じて、実・仮想ユーザの区別なく同じように空間が観測できること、(3) 各ユーザは空間内へのオブジェクトの導入や、各オブジェクトに対しアクション（移動や加工）を起こすことができ、かつ、そのリアクションが他のユーザに観測できること、が望ましい。また、普及性のためには、(4) 特殊な装置や高性能・高価なサーバやネットワークによらず上記の要件 (1)-(3) が実現できること、(5) 多数のオブジェクト（ユーザを含む）が同じ空間に同時に存在できること、も望まれる。

これまでに MMOG (Massively Multi-player Online Game) のための通信アーキテクチャ⁶⁾⁻⁸⁾ や遠隔協調作業のための様々な NVE^{9),10)} が提案されている。既存の MMOG アーキテクチャや NVE には、多数のユーザによる仮想空間の共有を P2P 技術を用いてスケー

^{†1} 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

^{†2} 広島市立大学情報化学研究科システム工学専攻
Hiroshima City University

^{†3} 滋賀大学経済学部情報管理学科
Department of Information Processing and Management, Shiga University

ラブルに実現しているものも存在し、要件 (1)–(5) を部分的に満たすが、仮想的な空間・オブジェクトの共有のみを対象としている。一方、既存の MR 技術や AR 技術を用いることで、実・仮想ユーザが、実・仮想オブジェクトを含む空間を同じように観測可能することができ、要件 (1)–(3) を満たすことは可能である。しかし、AR, MR 技術を実現するには、現状では特殊な装置や特別なサーバ、ネットワークを必要とするため、要件 (4), (5) を満たすことは難しい。

本論文では、実・仮想ユーザが同時に参加できる様々な対話型アプリケーションを、普及したコンピューティング環境で実現することを目的に、要件 (1)–(5) を満たすフレームワーク FAIRVIEW (Framework for Interaction between Virtual and Real Worlds)¹¹⁾ を提案する。FAIRVIEW では、要件 (1)–(3) を満たすため、アプリケーションの舞台となる実空間と、実空間に対応する仮想空間を用意し、両方の空間を重ね合わせる(以下、ハイブリッド空間と呼ぶ)。その空間では、実ユーザは仮想オブジェクトとその動作を、仮想ユーザは実空間の実オブジェクトとその動作を、それぞれ観測できる仕組みを実現する。要件 (4) を満たすため、FAIRVIEW の実行環境として、仮想ユーザはインターネット接続可能な PC のみを使用し、実ユーザは、無線通信機能を持つウェアラブルコンピュータ (HMD 含む) もしくは携帯端末を所持し、無線 AP (アクセスポイント) 経由でインターネット上の仮想ユーザの PC と通信できる環境を想定する。実オブジェクトの空間における位置、向きなどの情報(以下、AR 情報と呼ぶ)は、既存の AR 技術を用いて短い周期で計測し、ユーザ端末間でリアルタイムに交換することで、各ユーザごとの空間の視界をそのユーザの端末のディスプレイに 3D グラフィックスを用いて再現する。要件 (5) に対し、多数のオブジェクトが空間に存在するときも実・仮想ユーザ間の協調作業のために十分な精度でオブジェクトの動きを更新するため、AR 情報を実時間でユーザに配送する機構(以下、AR 情報配送機構と呼ぶ)を提案する。AR 情報配送機構には、ユーザノード間で利用可能な帯域の範囲内で、オブジェクトとそれを観測するユーザのペアごとに AR 情報の送信間隔を制御する QoS 適応機構を組み込む。この QoS 適応機構は、ユーザの視野およびオブジェクトとの距離に応じて、どのオブジェクトがユーザにとってより重要であるかを自動判定することでオブジェクト間の重み付けを行い、より重要なオブジェクトの動作をより滑らかに表示できるよう、AR 情報の送信間隔を決定する。

提案する QoS 適応機構について評価するため、文献 11) での各オブジェクトのデータの更新の頻度(以後、更新レートと呼ぶ)の試算およびそれをもとにした動画のユーザ満足度に関するアンケートの評価結果に加え、プロトタイプシステムを作成し、ユーザが視界を移

動した際に視界のオブジェクトの表示品質が適切に調整されるまでの時間をいろいろなオブジェクト数に対して計測した。その結果、通常のインターネット環境および無線 LAN 環境において提案機構が実行可能であることを確認した。また、仮想ユーザに対しては、実用上十分な更新レートと短い遅延を達成することを確認した。

2. 関連研究

文献 6)–8) は、ネットワークゲームにおける AOI (Area of Interest)^{*1} の管理手法に関する研究である。これらの研究は、仮想空間上で各ユーザが観測できる領域を様々な方法で限定し、限定された領域内のユーザ間でのみ空間の状態更新のための情報を交換することで、通信量を抑え、ユーザ数に対するスケーラビリティを保つ。文献 6) は、プレイヤーキャラクタの位置関係からゲーム空間をボロノイ図に基づいて動的に分割し、同じ領域に所属するプレイヤー間で直接イベントの交換を行わせることでゲーム状態を管理する手法を提案している。文献 7) は、空間をマイクロセルと呼ばれる小領域に分割し、各セルのプレイヤー数をもとに、複数あるサーバのそれぞれが管理するゲーム空間の範囲を動的に変更することで、ゲームのイベント処理の負荷をサーバ間で振り分ける方式を提案している。文献 8) では、共有仮想空間を八ニカム(六角形)構造で区切り、各八ニカムセルにおいて、各プレイヤーが自身の AOI 内に存在する他のプレイヤーやオブジェクトと効率良く情報交換する仕組みを、分散ハッシュテーブル(DHT: Distributed Hash Table)の 1 つである Pastry¹²⁾ を用いて実現するための手法を提案している。これら既存の NVE では、交換すべき情報をインターネットなどの普及ネットワーク環境でリアルタイムに交換できるようにするため、ユーザの観測可能エリアを制限している。しかし、本論文で対象とする実ユーザ・仮想ユーザ混合対話型アプリケーションでは、実ユーザや実オブジェクトの動きを仮想ユーザが滑らかに観測できる必要があり、そのために交換すべき情報量は、仮想空間のみを対象としたゲームなどの NVE に比べ、はるかに大きくなり、これら既存手法をそのまま適用するのは難しい。

文献 9), 10) は DVE (Distributed Virtual Environment) における負荷分散および QoS 適応制御に関する研究である。文献 10) では、共有空間を分割管理する既存ゲームアーキテクチャ^{6)–8)} で発生する「境界問題」(隣り合った空間が別のサーバで管理されることによる不整合など)に対処するため、共有空間全体を一括管理する専用サーバを複数設け、共有空

*1 ゲーム空間全体の中で、ゲームプレイヤーが関心を持っている部分空間のこと。一般には、そのプレイヤーが観測することのできる視野範囲が相当する。

間での処理を臨機応変に分散する方法を提案している．この手法は高性能サーバおよび高速ネットワークを前提とし、また、実空間を対象としていない．したがって、普及環境で QoS 適応制御を行うことで実ユーザ・仮想ユーザのインタラクションを実現する FAIRVIEW の目的とは異なる．文献 9) では、多数の遠隔ユーザ間で 3D 仮想空間を共有する大規模 DVE を実現するためのネットワークアーキテクチャ VESIR-6 を提案している．VESIR-6 では、ネットワーク資源の効率的な使用のために、共有オブジェクトの状態更新の配送に multicast を使用し、anycast を用いた処理の負荷分散を行うとともに、IntServ/DiffServ ベースの QoS 保証機構を用いてフローごとの伝送レートの制御を行っている．しかし、VESIR-6 は、実ユーザ・仮想ユーザ間の空間共有が必要となる無線ネットワークでの通信を考慮しておらず、また、オブジェクトの更新はマルチキャストグループへの join/leave により管理されているのみで、ネットワークリソースの制限に合わせて更新頻度を変更するという制御を行っていない．

地理的に離れた現実空間を仮想的に統合する環境を構築するための研究に、Tele-immersion (臨場感通信) がある．TEEVE¹³⁾ は、遠隔地のユーザに対して、各ユーザの視点で互いの 3D 実写映像を実時間で再現し、協調作業環境を実現している．しかし、Tele-immersion 環境を構築するには 3D カメラや広帯域のネットワーク環境など特殊かつ高価な装置や設備を必要とし、普及性の面で問題がある．

3. FAIRVIEW の概要

本章では、FAIRVIEW が提供する機能とアプリケーション例を示し、FAIRVIEW が想定するコンピューティング・ネットワーク環境と、実現のための基本方針を述べる．

3.1 FAIRVIEW が提供する基本機能

FAIRVIEW は、図 1 に示すように、実空間と仮想空間を重ね合わせ、実空間のユーザと仮想空間のユーザが現実に近い感覚で協調作業できる環境を提供する^{*1}．FAIRVIEW が提供する主な機能を以下に示す．

機能 (1) ユーザの視野に応じた空間の観測：仮想ユーザは、FPS (First Person Shooter) ゲームの要領で、マウスやキーボードを用いて自分の分身であるアバタを操作し仮想空間内を移動する．実ユーザは実空間内を普通に移動できる．その際、ユーザの位置、向きに応じ

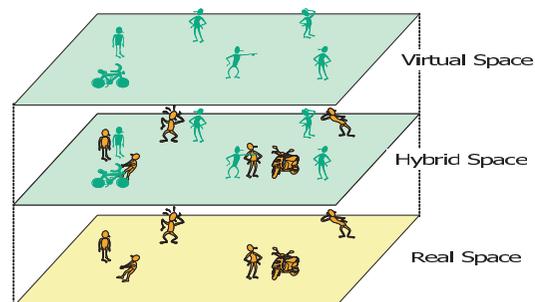


図 1 FAIRVIEW におけるハイブリッド空間
Fig.1 Hybrid Space Produced by FAIRVIEW.

て、仮想世界と現実世界の両方のオブジェクトがユーザにより観測される．

機能 (2) ユーザ間の音声会話：ユーザ間で位置関係に基づいた音声会話を可能とする．具体的には、近いユーザの声ほど大きく聞こえ、左側にいるユーザの音声は左側から聞こえるというような臨場感を提供する．この機能は、たとえば、文献 14) の手法を用いることで実現できる．

機能 (3) オブジェクトの共有：実オブジェクトと仮想オブジェクトをハイブリッド空間に登録し、実ユーザ・仮想ユーザの双方に観測ができるようにする．また、登録済みオブジェクトは登録を抹消し、見えなくすることもできる．共有オブジェクトは、そのオブジェクトが視野に入っているすべてのユーザにより観測される^{*2}．登録するオブジェクトの形状データはあらかじめシステムに登録されているものとする．

機能 (4) 登録したオブジェクトの移動・複数オブジェクトの結合：登録したオブジェクトに対して、近隣ユーザはアクションを起こすことができる．実オブジェクトは、実ユーザのみがアクションを起こすことができ、仮想オブジェクトは、実ユーザおよび仮想ユーザの両方がアクションを起こすことができる．たとえば、オブジェクトを移動させると、その動きが他のユーザに観測される．アクションとして、掴む、押す、引っ張る、回す、などを対象とする．これらのアクションは、オブジェクトを登録したユーザが管理責任を負う．管理責任を持つユーザの端末は、そのオブジェクトの状態の変化を管理する．また、複数のオブジェクトを位置関係を指定して結合することができる．あるオブジェクトを移動させると、

*1 FAIRVIEW は実空間のオブジェクトを仮想ユーザに見せることができるため、離れた複数の実空間を重ね合わせた視界をそれらの空間の実ユーザに提供することもできる．説明の簡単のため、以後は、実空間と仮想空間の間でのインタラクションに焦点を当てて説明する．

*2 実オブジェクトは、登録しなくても実ユーザに見えてしまうが、便宜上、空間に存在しないものとして扱う．

それと結合したオブジェクトと一緒に移動する。仮想オブジェクトと実オブジェクトを結合させた場合、実オブジェクトを移動させたときは仮想オブジェクトが連動して移動するが、仮想オブジェクトを移動させると、結合関係が解除される。

3.2 FAIRVIEW のアプリケーション

本節では、多数のユーザ間の実時間インタラクションが重要であるアプリケーションの例を 2 種類あげ、特に必要とされる機能と各アプリケーションに特有の達成すべき要件について説明する。

蚤の市タイプ このアプリケーションでは、実空間に存在する蚤の市に、実ユーザに加えて、仮想ユーザが、売り手・買い手として参加する。たとえば、実ユーザである売り手は商品（ここでは実オブジェクト）をハイブリッド空間に登録する。仮想ユーザの買い手は、蚤の市の会場を歩きながら、どんな商品をどんな人が売っているのかを視覚的に概観することができる。また、売り場では、実ユーザと仮想ユーザがお互いの存在を認識しながら音声で会話することができる。さらに、売り手は、商品を移動・回転させるなど、商品の詳細を買い手に見せながら、インタラクティブに商品の説明を行うことができる。以上は、仮想ユーザが売り手で、実ユーザが買い手の場合も実現できる。展示会や見本市、ショッピングセンタなどのアプリケーションがこのタイプに属し、FAIRVIEW により実現できる。このタイプのアプリケーションは、商品・展示品に関する共通の認識によるコミュニケーションによって成り立つため、機能 (3), (4) が重要である。また、機能 (1), (2) を採用することによって、よりスムーズなコミュニケーションを実現できる。さらに、これらのアプリケーションでは、多くの商品・展示品を表現する必要があるため、多数のオブジェクトを扱えることは実現する際の重要な要件となる。

対戦ゲームタイプ 実空間でのサバイバルゲームに、仮想ユーザが参加する。典型的なサバイバルゲームでは、銃および的を持ったプレイヤーが互いに対戦相手の的を撃ち、的を撃たれたプレイヤーが脱落していくなか、最後まで残ったプレイヤーが勝利をおさめる。FAIRVIEW において、各プレイヤーは自分自身のアバタおよび銃、的を共有するだけで、他のプレイヤーが実ユーザか仮想ユーザかを意識することなく、互いの存在および位置関係を認識し、協力したり、戦ったりすることができる。FAIRVIEW を用いることで、実空間では登場させることができない敵キャラクタ、たとえば巨大な恐竜などを登場させることが可能になる。また、実ユーザだけでは参加者が少ない場合に、参加者数を容易に補うとともに、仮想ユーザには、より現実に近いプレイ感覚を与えることができる。街角でのイベントや、テーマパークのアトラクションなどのアプリケーションがこのタイプに属し、FAIRVIEW により

表 1 ユーザの装備
Table 1 User equipment.

Type	Computer	Display	Network	Other
Real User	PC / PDA	HMD / etc	Wi-Fi / etc	ウェブカメラ, センサ
Virtual User	PC	LCD / etc	インターネット	マウス / etc

実現できる。このタイプのアプリケーションは、状況の変化によるインタラクションによって成り立つため、機能 (1), (3) が重要である。また、機能 (2), (4) を採用することによって、アプリケーションのエンタテインメント性を向上させることができる。さらに、これらのアプリケーションでは、状況の変化がユーザの行動に大きな影響を及ぼすため、ユーザの行動をはじめとしたイベントに対する即応性を持つことが求められる。

3.3 FAIRVIEW に必要なデバイスとネットワーク

表 1 にユーザが必要な装備を示す。実ユーザは、無線 LAN を備えた小型計算機 (PDA など) と HMD (ヘッドマウントディスプレイ) を用いる。そのほかに、実空間中の位置や向いている方向を計測するセンサ (GPS, トラッキング装置など) およびヘッドセットなどの音声入出力機器を備えるものとする。HMD やセンサは、Wii や PS3 などのゲームコンソール向けの安価なものを流用することを想定する。仮想ユーザは、PC, LCD などのディスプレイ装置、マウスなどのポインティングデバイス、音声入出力装置を備えるものとする。

3.4 FAIRVIEW 実現のための基本アイデア

AR 技術を用いるアプリケーションにおいて、ユーザが観測する視界をどのように生成し描画するかは、コストなどの面で重要な問題である。TEEVE¹³⁾ では、3D マルチカメラによりキャプチャした映像に高度な画像処理を適用し、広帯域ネットワークである Internet2 を経由して伝送する。FAIRVIEW は普及環境での実現が目的なので、各オブジェクトの AR 情報をセンサにより計測し、ユーザ端末間でリアルタイムに交換し、各端末で AR 情報をもとにオブジェクトを 3D 描画する方法を採用する。本方式に関して、解決すべき問題は以下の 3 つである。

- (i) 実オブジェクトに登録し、それを移動させた際に、空間に対するそのオブジェクトの AR 情報を正確に計測すること
- (ii) ユーザ端末で、オブジェクトおよびその動作を違和感なく描画すること
- (iii) 普及ネットワーク環境で、AR 情報をリアルタイムにユーザ端末に届けること

上記 (i) の実現において、位置の計測は、屋外では GPS が利用でき、屋内では、無線 LAN の AP を用いた手法¹⁵⁾ や音源とマイクを用いた手法¹⁶⁾、Place Lab¹⁷⁾、Weavy¹⁸⁾ などが

利用できる。また、オブジェクトの向きや傾きについては、姿勢推定機器によって計測するか、ARToolkit¹⁾ や、文献 2) で提案されている画像処理ベース計測手法を用いることができる。

上記 (ii) については、3D グラフィックスをリアルタイムに描画できる性能を持ったユーザ端末を用いる^{*1}。

上記 (iii) は本論文が扱うメインピックであり、その実現のためには、無線ネットワーク上の実ユーザの端末とインターネット上の仮想ユーザの端末がリアルタイムに AR 情報を交換できる通信機構 (AR 情報配送機構) が必要である。3.2 節で述べた蚤の市や対戦ゲームのようにハイブリッド空間に多数のオブジェクトが存在するアプリケーションでは、単位時間あたりに交換すべき AR 情報の総量が大きくなり、利用可能帯域内に収まらなくなる。そのため、単位時間あたりに AR 情報を計測・配送する頻度を下げるなどの方策が必要である。しかし、FAIRVIEW が対象とするインタラクティブなアプリケーションでは、AR 情報の計測・配送頻度を極端に下げると、インタラクションの効果が大きく損なわれる。そこで、ユーザ・オブジェクトの位置関係、向きなどに応じて、オブジェクト間で AR 情報の更新頻度を重み付けし、制約のある中で、できるだけユーザ間のインタラクションに関するユーザの満足度が高くなるような QoS 適応機構を考案する。

4. AR 情報配送機構

本章では、AR 情報配送機構について述べる。

4.1 諸定義

主な諸定義を以下に示し、表 2 にまとめる。 R を実空間、 V を仮想空間とする。 $H = (R, V)$ を R と V を重ねることによって生成されるハイブリッド空間とする。ハイブリッド空間 H は 3D 座標系の x - y 軸で並べられた直方体であると仮定する。 $RO = \{ro_1, \dots, ro_n\}$ 、 $VO = \{vo_1, \dots, vo_m\}$ 、 $RU = \{ru_1, \dots, ru_l\}$ 、 $VU = \{vu_1, \dots, vu_k\}$ をそれぞれ、 R の実オブジェクトの集合、 V の仮想オブジェクトの集合、実ユーザの集合と仮想ユーザの集合とする。また、ユーザを含めた全オブジェクトの集合を AO とする。 $node(u)$ を、ユーザ $u \in RU \cup VU$ のユーザ端末とする。このとき、 $RN = \{node(u)|u \in RU\}$ および $VN = \{node(u)|u \in VU\}$ は、それぞれ実ユーザ端末の集合と仮想ユーザ端末の集合である。さらに、オブジェクト $ao \in AO$ の AR 情報を $AR(ao) = (pos, angle)$ と表記する。こ

表 2 諸定義
Table 2 Definitions.

	表記	定義
実空間	R	
仮想空間	V	
ハイブリッド空間	H	$H = (R, V)$
実オブジェクト	RO	$RO = \{ro_1, \dots, ro_n\}$
仮想オブジェクト	VO	$VO = \{vo_1, \dots, vo_m\}$
オブジェクト	o	$o \in RO \cup VO$
実ユーザ	RU	$RU = \{ru_1, \dots, ru_l\}$
仮想ユーザ	VU	$VU = \{vu_1, \dots, vu_k\}$
ユーザ	u	$u \in RU \cup VU$
ユーザ端末	$node(u)$	
実ユーザ端末	RN	$RN = \{node(u) u \in RU\}$
仮想ユーザ端末	VN	$VN = \{node(u) u \in VU\}$
全オブジェクト	AO	$AO = RO \cup VO \cup RU \cup VU$
ユーザを含むオブジェクト	ao	$ao \in AO$
AR 情報	$AR(ao)$	$AR(ao) = \{pos, angle\}$
属性情報	$attribute$	$attribute = \langle color, form, \dots \rangle$
アクション	$action$	$action = \langle type, direction, strength, \dots \rangle, type \in \{push, pull, \dots\}$
AR イベント	$ARe(ao)$	$ARe(ao) = (AR(ao), attribute, action)$

こで、 $ao.pos$ は ao の位置、 $ao.angle$ は ao の各座標軸に対応する回転角度である。また、AR 情報に色、形状などの属性情報 $attribute = \langle color, form, \dots \rangle$ や他のオブジェクトへのアクション $action = \langle type, direction, strength, \dots \rangle$ 、 $type \in \{push, pull, \dots\}$ を付加したものを AR イベント $ARe(ao) = (AR(ao), attribute, action)$ と表記する。

本論文では、各実ユーザ ru が 3 章で説明した装備を持ち、そのユーザの AR 情報を 1 秒あたり 60 回測ることができ、また、ARToolkit とウェブカメラなどの技術によって、各実オブジェクト ro の AR 情報を測定できると仮定する。

4.2 ユーザ同士のコミュニケーションにおける仮定

簡単のため、実空間 R 全体がインターネットに接続している 1 つの AP によってカバーされると仮定する。 BW_{AP} を実ユーザ端末と AP の間で利用できる帯域幅であるとする。このとき、すべての実ユーザ端末がこの帯域幅を共有することになる。各仮想ユーザ端末 vn に関して、 $bw_{AP}(vn)$ を vn と AP の間で利用できる帯域幅であるとする。各実ユーザ端末 rn について、 $bw(vn, rn)$ を vn と rn の間で利用できる帯域幅とする。このとき、

*1 近年、3D グラフィックス描画機能を持った携帯電話端末や PDA は標準的であり、普及している。

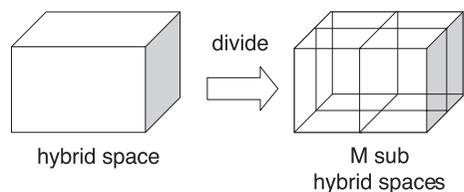


図2 ハイブリッド空間の分割
Fig. 2 Division of hybrid space.

$bw(vn, rn) = \text{Min}(BW_{AP}, bw_{AP}(vn))$ である。また、仮想ユーザ端末 $vn1, vn2$ に対して、 $bw(vn1, vn2)$ を $vn1$ と $vn2$ の間で利用できる帯域幅とする。

4.3 AR イベント配送機構

ユーザの AOI に基づき、AR イベントの配送処理・配送量のノードあたりの負荷を減らすため、図2に示すようにハイブリッド空間 H を小さな直方体の部分領域に分け、各部分領域にエリアノード (area node) と呼ばれるサーバノードを割り当てる。これは、文献19) のような既存の P2P ベースの MMOG ゲームアーキテクチャと同様のアプローチである。 an_A を部分領域 A に割り当てられるエリアノードとする。エリアノード an_A は、 A 内のオブジェクトの AR イベントを受信すると、 A や近隣の部分領域でそのオブジェクトを観測しているユーザにその AR イベントを配送する。FAIRVIEW は、オブジェクトの移動に加え、向きや傾きなどの細かな動きも再現することを目指しているため、処理・伝送する情報量が MMOG に比べ多い。また、ネットワークの資源の制約に合わせて各ユーザに配送する情報を調整する必要がある。そのため、エリアノードに加え、帯域制御ノード (bwc-node) を導入し、ハイブリッド空間の状態の維持・管理を、これらのノード群に分散処理させる。帯域制御ノードは、各仮想ユーザ端末 vn および AP に対し用意される^{*1}。 $bn(u)$ をユーザ u のための帯域制御ノードとする。 $bn(u)$ は、 $bw(bn(u), node(u))$ をモニタし、 an_A から $node(u)$ への AR イベントのストリームを QoS 適応制御する。図3に、FAIRVIEW によるオーバーレイネットワークの概要図を示す。これは1つの部分領域に対応している。この部分領域に属するエリアノードを1個、ユーザノードを N 個、帯域制御ノードを M 個とする。以下、この図を用いて AR イベントの配送手順を説明する。

*1 FAIRVIEW では、無線帯域幅の制限のため、それぞれのオブジェクト ao をすべての実ユーザが同じ更新レートで観測するように制限する。

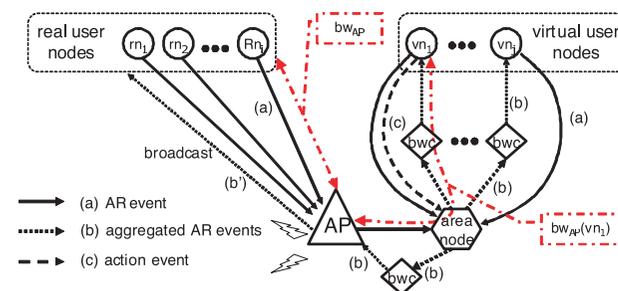


図3 オーバーレイネットワーク
Fig. 3 Overlay network for AR event delivery.

FAIRVIEW は、次のような機能を持つ：(1) ユーザ u は、自身も他のユーザに観測される、(2) u は、その視界にある他のオブジェクト ao を見ることができる、(3) u は、他のオブジェクト ao にアクションを起こすことができる。

上記(1)のために、 $node(u)$ は連続的に u の AR 情報を測っており、それが以前の測定値と異なるならば、 $node(u)$ はそれを AR イベントとしてエリアノード an_A に送る (図3(a))。上記(2)のために、 $node(u)$ は、 u の視界にあるオブジェクト ao の AR イベントを受信し、オブジェクトの最新の状態を $node(u)$ のディスプレイに表示する。 $node(u)$ が AR イベントを受信するために、publish/subscribe 方式²⁰⁾を用いる。 $node(u)$ が、エリアノード an_A に AR イベントを送ったとき、 an_A は AR イベントによって u の視界のオブジェクト ao を確認する。そして、 an_A は帯域制御ノードを通して、 $node(u)$ にオブジェクト ao の AR イベントを配送する (図3(b), (b'))。図3(b')に示すように、実ユーザ端末は、実空間 R に配置された AP からのブロードキャストにより、帯域制御ノードから AR イベントを受信する。上記(3)のために、 u が仮想オブジェクト o にアクションを起こすとき、 $node(u)$ は an_A に、力の強さ・向きと動作が記述されたアクションを含む AR イベントを送信する (図3(c))。

全体の処理の流れは以下ようになる。

Step(1) ユーザノード $node(u)$ は、AR イベント $ARe(u)$ をエリアノード an に向けて送信する。このとき、ユーザノード $node(u)$ は、オブジェクト o の管理責任を持つなら、オブジェクト o の AR 情報を取得し、AR イベント $ARe(o)$ を作成し、エリアノード an に向けて送信する。

Step(2) エリアノード an は、5章で述べる QoS 制御手法に従い、AR イベント $ARe(ao)$

に重要度を付加する．

Step(3) エリアノード an は、重要度を付加した AR イベント $ARe(ao)$ を帯域制御ノード bn に送信する．

Step(4) 帯域制御ノード bn は、定期的に 5 章の QoS 制御手法に従い、重要度を用いて帯域 $bw(node(u), bn)$ の再分配を行い、それぞれの AR イベント $ARe(ao)$ に対して更新レートを決定する．

Step(5) 帯域制御ノード bn は、Step(4) で決められた更新レートに従い、送信する AR イベント $ARe(ao)$ をまとめ、ユーザノード $node(u)$ に送信する．

Step(6) ユーザノード $node(u)$ は受信した AR イベント $ARe(ao)$ をもとに、仮想空間のデータを更新する．

Step(7) ユーザノード $node(u)$ は、更新されたデータをもとに、周期的に仮想空間の描画を行う．

Step(8) Step(1) に戻る．

5. 視界に基づく QoS 適応制御

通信量を利用できる帯域幅以内に抑える手法として、不必要だと思われるデータを推定し送信しない方法や情報の精度を下げデータサイズを小さくする方法が考えられるが、本論文では、各ユーザがより注目したいオブジェクトを、高品質で見せるための制御をフレキシブルに行うために、以下の手法を採用する．QoS 適応機構の基本アイデアは以下のとおりである：(1) オブジェクトがユーザにとってどれくらい重要かについて、各オブジェクトの相対的な重要度を定める．そして、(2) 各ユーザのために、通信量の合計が利用できる帯域幅より小さくなるように、重要度に基づいて観察しているオブジェクトの AR イベントの通信量を管理する． $Watcher(o)$ を、オブジェクト o を観察できるユーザの集合とする．このとき、 $Watcher(o)$ は、以下のように定義できる．

$$Watcher(o) \stackrel{def}{=} \begin{cases} \{u | u \in RO \cup VO, o.pos \in View(u)\} & (o \in VO) \\ \{u | u \in VU, o.pos \in View(u)\} & (o \in RO) \end{cases}$$

5.1 オブジェクトの重要度の決定

ここで、 $View(u)$ は、ユーザ u のハイブリッド空間 H における視野範囲 (図 4 の半円全体) を表し、 $o.pos$ は o の H 上の位置を表す．各ユーザ $u \in Watcher(o)$ について、 o が u にとってどの程度重要かを表す値を重要度と呼び、 $Imp(o, u)$ と表記する．このとき、

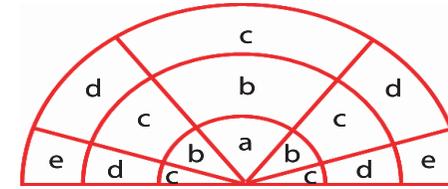


図 4 ユーザの視界と領域ごとの重要度

Fig. 4 User's view and zones with relative importance.

$View(u)$ の形状および重要度の割当ては、サービス提供者がアプリケーションごとに最適なものにカスタマイズするのが理想であるが、本論文では、1 つの例として、扇型の視野範囲を設定し、近くのオブジェクトほど、また、視野範囲の中央のオブジェクトほどより詳細に観察できるような視野の分割を用いて、重要度の割当てを行う．現実世界の場合にならうと、重要度 $Imp(o, u)$ は、ユーザ u とオブジェクト o の距離が近いほど、また o が u の視野の中央にあるほど高いのが自然である．視野範囲外のオブジェクトの重要度は 0 にする．すなわち、重要度は、視野におけるオブジェクトの距離と位置により決定する．そのため、提案手法では、図 4 に示すように、ユーザの視野範囲を a-e の 5 つの部分領域からなる半円で表し、部分視野範囲 a に存在するオブジェクトの重要度は最も高く、b, c, d, e の順に、それらの部分視野範囲に存在するオブジェクトの重要度は低くなる．つまり、アルファベット順に小さくなっていき、半円の外の領域では 0 となる．

5.2 AR 情報配送の更新レートの決定

各オブジェクトからユーザ u に対して配信される AR イベントの通信量を、 u の視界のすべてのオブジェクトの重要度の合計に対するその重要度の比率に基づいて決定する．このとき、 $bw(bn(u), node(u))$ の通信量に収まるように各オブジェクトの更新レートが削られる．提案手法では、更新レートを削るために、帯域制御ノードにおける AR イベントのバケットドロップを行う．以下では、オブジェクトの重要度に基づいた QoS 適応制御の例について、AR 情報を受け取るユーザが、仮想ユーザである場合と、実ユーザである場合に分けて説明する．

5.2.1 仮想ユーザに対する QoS 適応

仮想ユーザ v の端末 $node(v)$ が、 v の視野範囲にある 3 つのオブジェクト o_1, o_2, o_3 の AR 情報を受け取る場合を想定する．AR 情報が v に対して割り当てられた帯域制御ノード bn_v を経由して、ユーザ端末 $node(v)$ に受信されるとする．なお、 bn_v と $node(v)$ の間で利

用可能な通信帯域は 1 Mbps であると仮定する．今, $on(o_1)$, $on(o_2)$, $on(o_3)$ が送信する AR 情報配送に必要な伝送速度がそれぞれ 0.5 Mbps, 総量は 1.5 Mbps とする．この場合, オブジェクト o_1 , o_2 , o_3 の重要度の比に従い利用可能帯域 1 Mbps を比例配分する． o_1 , o_2 , o_3 の重要度がそれぞれ 10, 25, 15 とすると, それぞれ, 0.2 Mbps, 0.5 Mbps, 0.3 Mbps の帯域が割り当てられる．この結果をもとに, それぞれ割り当てられた帯域と AR イベントの平均データサイズから 1 秒間に送れるパケット数を算出する．このとき, 帯域制御ノード bn がユーザノード $node(u)$ に対し, 1 秒間に送るパケット数を更新レートとする．更新レートは,

$$AR \text{ 情報の計測回数} \geq \text{更新レート} \geq 0$$

である．この更新レートをもとに, 帯域制御ノード bn で, パケットドロップを行うことで, 各オブジェクトの AR イベントパケットの通信量を調整する．

5.2.2 実ユーザに対する QoS 適応

実ユーザ r の端末 $node(r)$ が, r の視野範囲にある場合, AR 情報が無線 AP に割り当てられた帯域制御ノード bn_W およびその無線 AP を経由して, ユーザ端末 $node(r)$ に受信される．このとき, 実ユーザ r に観測される仮想オブジェクト o の重要度 $Imp_r(o)$ を, 以下のように定義する:

$$Imp_r(o) = \text{Max}_{r \in \text{Watcher}(o)}(Imp(o, r))$$

このように定義するのは, 実ユーザらの重要度は無線 AP (およびその帯域制御ノード) によってまとめられるため, r の重要度は o を観察している実ユーザ全員の重要度における最大値と同じになるためである．帯域制御ノード bn_W は, 実ユーザの AR イベントの通信の際に QoS 適応制御をする．各オブジェクトの更新レートは, 5.2.1 項と同じように, 帯域を比例配分することで決定する．

6. 実験

本論文で提案したフレームワークの有効性を評価するため, オブジェクト数に対するトラフィック量およびユーザが見るオブジェクトの更新レートをシミュレーションにより計測した．また, 作成したプロトタイプシステムを用いて, 重要度の算出および重要度による QoS 適応制御を行った場合にかかる処理遅延, AR イベントを送信し, 受信するまでのエンド・エンド遅延を, LAN ネットワーク上で計測した．

6.1 実験環境

各実験における, 設定を表 3 に記す．フィールドの大きさは, 運動場を想定し, 50×50 [m²]

表 3 実験環境

Table 3 Experimental configurations.

設定項目		実験 6.2, 6.3	実験 6.4
ユーザ数	実	100	—
	仮想	100	1
ハイブリッド空間の広さ (m ²)		50×50	6×6–120×120
帯域 (Mbps)	有線	10	—
	無線	5	5
仮想オブジェクト数		50–5,000	8–1,683
パケットサイズ		32	32–64
最大更新レート (frame/sec)		60	30

とする．また, 図 4 のように, ユーザの視界は, 各角度が $\pi/3$, 各領域の距離が 5 m となるように, 15 m の半円を分割した．これは, 全体のオブジェクト数が 50 個のとき, 視野に入るオブジェクトが, 1 つから 2 つ, かつ, 整数になるように設定している．また, 領域 a, b, c, d, e の重要度は, それぞれ, 64, 16, 4, 2, 1 とした．これは, 図 4 において, 領域 c の面積が最も大きいため, 領域 e から, それぞれ 2 倍ごとに設定すると, 領域 c が領域 a, b より, オブジェクト数で上回ってしまうため, 効果的に重要度の効果を得ることができない．よって, 領域 c を基準に, 領域 c より遠い領域の重要度は 1/2 倍ごと, 領域 c より近い領域の重要度は 4 倍ごとに重要度を設定した．特別な記述がない限りは, この設定を用いる．これらの実験環境は, 実験 6.2, 6.3 では, 多数のオブジェクトを配置したときの使用帯域, 更新レートを計測するため, 蚤の市アプリケーション全体を模した環境になっており, 実験 6.4 では, ユーザが動き回るときの QoS 適応制御が反映されるまでの遅延, パケットが配送されるまでのエンド・エンド遅延を計測するため, ゲームなどのアプリケーションにおいて, 参加したユーザに注目した場合の実験環境になっている．

6.2 ユーザ端末のトラフィック

提案手法のスケラビリティを評価するために, QoS 適応制御機構がない場合とある場合の両方について, オブジェクトの数を変化させて, ユーザ端末とそれに対応する帯域制御ノードの間で必要なネットワークの使用帯域幅を計測した．シミュレーションは表 3 の設定で行った．また, そのほかの設定は以下のとおりである．仮想空間と同じ大きさの実空間が存在する．AP は 1 つとし, AP の伝播範囲は実空間 R 全体をカバーする．無線帯域 BW_{AP} は, すべての実ユーザがこの帯域を共有する．エリアノード, オブジェクトノード, 帯域制御ノードは有線接続の固定サーバ上に割り当てられる．すべてのオブジェクトは, 八

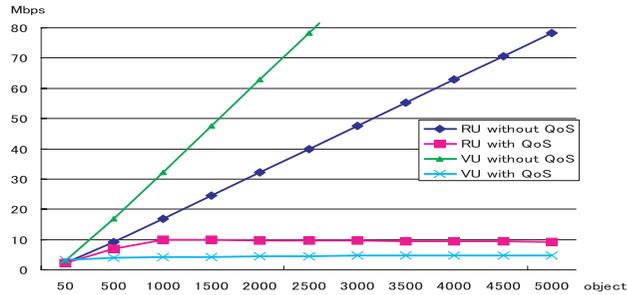


図 5 ユーザ端末にかかるトラフィック
Fig. 5 Required bandwidth for user terminals.

イブリッド空間に無作為に配置される．ユーザの方向もランダムに決められる．実験 6.2, 6.3 では，各ユーザ端末は (X, Y, Z, angleX, angleY, angleZ, オブジェクト名, 重要度 (各 4 [byte])) = 32 [byte] からなる AR イベントを毎秒 60 個送信する．

実験結果を図 5 に示す．結果は 100 試行の平均である．QoS 適応制御を行わない場合，オブジェクトの数がそれぞれ，250, 500 以上のとき，ユーザ端末のトラフィック量は，各実ユーザ端末（図 5 中 “RU without QoS”）と各仮想ユーザ端末（図 5 中 “VU without QoS”）に必要な帯域幅は限界（すなわち 10 Mbps と 5 Mbps）を上回った．一方，QoS 適応制御を行う場合，たとえオブジェクト数が 5,000 を超える場合でも，必要な帯域幅は限界を超えないように管理することができた（図 5 中 “RU with QoS” および “VU with QoS”）．以上のことから，ユーザ数とオブジェクト数が大きい場合には，QoS 適応機構が有効に働くと考えられる．これにより，蚤の市などの多数のユーザやオブジェクトが存在するアプリケーションでも利用帯域を抑えることができ，本フレームワークは有効であることが分かる．

6.3 QoS 適応制御の効果

提案手法では，より重要なオブジェクトの AR イベントは，利用できる帯域幅の範囲内で，他のオブジェクトより大きい通信量で送られる．QoS 適応制御の影響を調べるため，6.2 節で述べた領域 a-e にあるオブジェクトの AR イベントの更新レートを計測した．この実験では，オブジェクトが増加したとき，ユーザにとって重要なオブジェクトが，QoS 適応制御により，どの程度まで更新レートを維持できるかを計測した．シミュレーションの設定は，実験 6.2 と同様である．

結果を，図 6 と図 7 に示す．図 6 と図 7 において，a, b, c, d, e のラベルは，それぞ

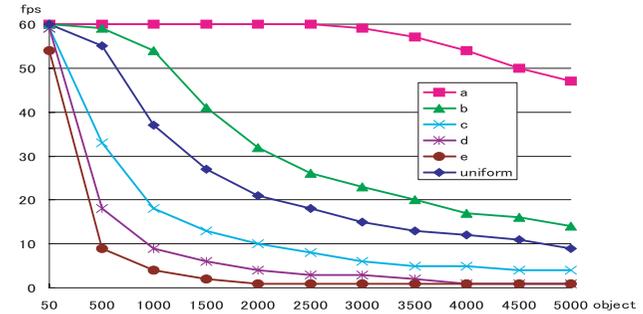


図 6 仮想ユーザが観測するオブジェクトの品質
Fig. 6 Framerates at virtual users.

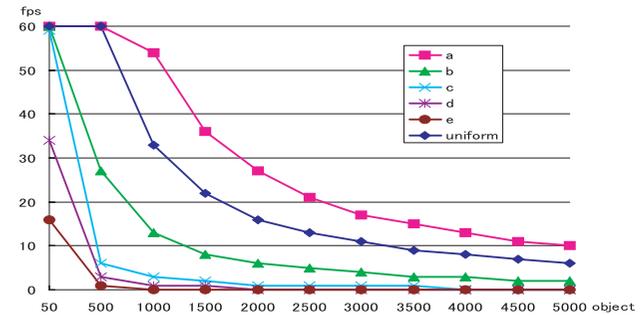


図 7 実ユーザが観測するオブジェクトの品質
Fig. 7 Framerates at real users.

れに対応する領域の平均更新レートを示す．また，uniform のラベルは，帯域幅が一様に分配される場合の平均更新レートを示す．図 6 より，各仮想ユーザが，領域 a, b の重要なオブジェクトを uniform より高い更新レートで見ることができることが分かる．特に，オブジェクト数が 4,500 以下では，領域 a のオブジェクトの更新レートは，50 fps を保っている．領域 c, d, e の重要でないオブジェクトの更新レートは，uniform より減らされている．

図 7 より，各実ユーザが領域 a の重要なオブジェクトを uniform より高い更新レートで見ることができることが分かる．しかし，その効果は仮想ユーザの場合よりも小さい．他の領域のオブジェクトの更新レートは，uniform より減らされている．これは，5.1 節で説明したように，実ユーザ全員における重要度の最大値を各実ユーザの重要度とするため，多く

のオブジェクトがユーザにとって重要なオブジェクトと見なされてしまうからである。しかし、オブジェクト数が 2,000 未満のとき、領域 a のオブジェクトの更新レートはおおいに改善されている。この実験結果により、一般的な扇形の視野と各範囲に固定された重要度による設定でも QoS 適応制御によって効果的にデータの更新レートを制御でき、本フレームワークは有効であることが分かる。任意のアプリケーションに特化した重要度の設定法を用いることで、アプリケーションに則した QoS 適応制御を実現することができる。

6.4 QoS 適応制御における処理遅延

提案する QoS 適応制御を行った場合にかかる遅延を評価するために、LAN 環境において、ユーザの視野が変化した場合に QoS 適応制御によって更新レートが変化するまでの遅延時間を計測した。実験は表 3 の設定で行った。また、そのほかの設定は以下のとおりである。ユーザは仮想ユーザ 1 人とし、実空間および AP は存在しないものとする。ユーザはまず、空間の中心に配置され、ランダムに決められた目的地に向かって、1 m/sec で前進する。目的地に到着すると、次の目的地が再びランダムに決められる。仮想空間において、 n 個の仮想オブジェクトがユーザを中心として 3 m 間隔に配置される。このとき、ハイブリッド空間は最も遠くに配置されたオブジェクトが入るだけの広さとする。実験 6.4 では、各ユーザ端末は、(位置情報 (8 [byte] × 3), 回転角度 (8 [byte] × 3), 重要度 (4 [byte]), ユーザ ID (4 [byte]), オブジェクト名 (4 [byte]), アクション (4 [byte])) = 64 [byte] からなる AR イベントを毎秒 30 個送信する。実験に用いたプロトタイプでは、AR イベントによる各オブジェクトのデータの更新、パケットの送信、パケットの受信、重要度の変更および QoS 適応制御は、1 秒間に 30 回の頻度で並列処理される。このとき、それぞれのプロセスは個別のタイムスロットを使い非同期で処理される。重要度の変更はエリアノードが、QoS 適応制御は帯域制御ノードが行う。通信プロトコルには TCP を使用した。実験には、エリアノード、帯域制御ノード、ユーザノードにそれぞれ、CPU: Athlon64 X2 4200+, メモリ: 2 GB, OS: Debian Linux (kernel 2.6.8), CPU: Athlon64 X2 4200+, メモリ: 2 GB, OS: Debian Linux (kernel 2.6.8), CPU: Opteron 242 (1.6 GHz) × 2, メモリ: 8 GB, OS: Debian Linux (kernel 2.6.8) のマシンを使用した。プログラム言語には、JAVA SE 1.6.0_05 を用いた。重要度の変更と QoS 適応制御に関する処理遅延は、計算処理を始める直前と終了した直後のタイムスタンプの差分によって計測した。また、エンド・エンド遅延は、ユーザノードが送信する直前とデータを更新した直後のタイムスタンプの差分によって計測した。これらに、スレッドの生成遅延は含まない。

実験結果を図 8, 図 9 に示す。結果は 5 試行の平均である。図 8 は、アバタの位置に従

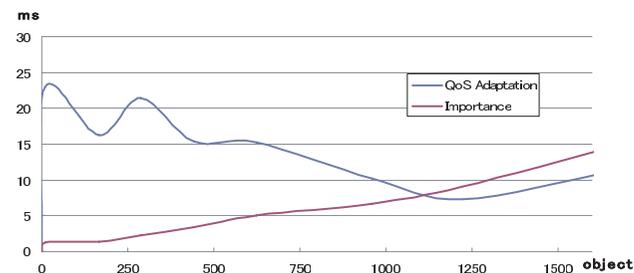


図 8 QoS 適応の遅延時間
Fig. 8 Delay by QoS adaptation.

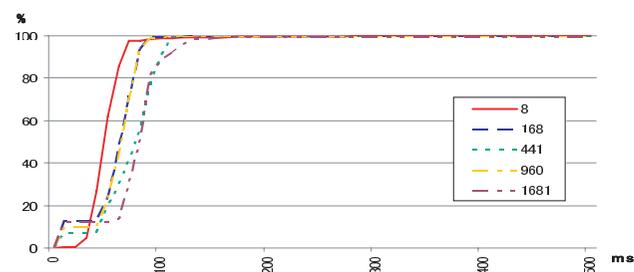


図 9 AR イベントが伝わるまでの処理遅延の累積分布
Fig. 9 Cumulated distribution of end-end delay.

い変化する重要度の再計算 (図 8 中 “importance”), および、それをもとにした QoS 適応制御にかかる再計算 (図 8 中 “QoS Adaptation”) の処理遅延のグラフである。これより、重要度の再計算はオブジェクト数に従い計算量が増加するため、処理遅延が大きくなることが分かる。この処理遅延の主な原因は、アバタ対オブジェクトの距離・角度計算である。これにより、オブジェクト数が多い環境では、短い周期 (たとえば、フレームごと) で計算するのは難しいことが分かる。このとき、CPU を効率的に使うために、再計算の頻度を下げるか、エリアノードの数を増やす必要がある。一方、QoS 適応制御にかかる時間は処理のタイミングによって振動するものの、25 ms 以下に収まっており、オブジェクト数に依存しないことが分かった。

さらに、重要度の再計算と QoS 適応制御にかかる再計算の合計により、ユーザが振り向くなどの急激なアクションへの耐性をみる事ができる。つまり、ユーザが振り向きなどの

動作によって、各オブジェクトの重要度が大幅に変化した場合、その影響がどの程度の遅延で反映されるかが分かる。図 9 は、図 8 の結果を含め、ユーザノードが AR イベントを送信し、その AR イベントがエリアノードに受理され、帯域制御ノードを経由して戻ってくるまでのエンド・エンド遅延を測ったものである。およそ 200 ms 以内に収まっている。遅延の主な原因は、重要度の再計算処理、QoS 適応制御にかかる再計算処理、非同期の計算プロセスによる待機時間が考えられるが、図 8 より、非同期による待機時間が最も大きな原因であることが分かる。図 8 および図 9 からオブジェクトが 1,500 個を超えた状況でも 30 ms 程度の計算遅延で処理され、その影響は遅くとも 200 ms 以内にはユーザに反映されることが分かる。人間が振り向き、対象にフォーカスをあわせる、という動作にかかる時間を考慮すれば、この遅延時間は、実用上、十分に許容範囲内である。これにより、本フレームワークは、即応性の求められるゲームなどにも、十分に対応できることが分かる。

7. ま と め

本論文では、多人数参加型の実・仮想空間のインタラクションを実現するためのフレームワーク FAIRVIEW を提案した。提案手法では、共有空間を分割したうえで、Publish/Subscribe 方式を用いることで、ユーザ自身を含む動くオブジェクトに関する情報の共有を行う。また、普及ネットワーク環境での利用を目的に、インタラクションの際のユーザ満足度を高く保ちながら通信量を削減するための QoS 適応機構を提案し、実験による評価を行った。実験の結果、ユーザの視界にあるオブジェクトの動きを十分に実用可能な範囲の更新レートに保ちつつ、通信量を利用可能な帯域に収めることができることが分かった。また、その際にかかる計算処理にかかる遅延時間も実用上、十分に許容範囲であることを確認した。

今後の課題は、以下の 4 つである。無線通信環境下では、インターネットに比べ、遅延が大きく変動する傾向がある。そのような環境下において、遅延を小さく抑えるような対策を考案する必要がある。また、無線環境下において、同じエリアでアクセスポイントが 2 つ以上ある場合について実験を行い、実ユーザのオブジェクトの観測性能の向上効果を調査する必要がある。さらに、提案した QoS 適応制御手法では、オブジェクトを一律に重要度で扱うことによって、ゲームにおいて、標的と障害物が同等に扱われてしまう。今後、オブジェクトの種類による重要度の調整も検討したい。最後に、提案手法をミドルウェアとして実装し、テストアプリケーションを構築して評価を行う必要がある。

参 考 文 献

- 1) ARtoolkit. <http://www.hitl.washington.edu/artoolkit/>
- 2) Fudono, K., Sato, T. and Yokoya, N.: Interactive 3-D modeling system using a hand-held video camera, *Proc. 14th Scandinavian Conference on Image Analysis (SCIA 2005)*, pp.1248–1258 (2005).
- 3) Ichikari, R., Kawano, K., Kimura, A., Shibata, F. and Tamura, H.: Mixed Reality Pre-visualization and Camera-Work Authoring in Filmmaking, *Proc. 5th International Symposium on Mixed and Augmented Reality*, pp.239–240 (2006).
- 4) Fujimoto, M. and Ishibashi, Y.: Packetization Interval of Haptic Media in Networked Virtual Environments, *Proc. 4th ACM Workshop on Network and System Support for Games (NETGAMES '05)* (2005).
- 5) Yang, Z., Yu, B., Nahrstedt, K. and Bajscy, R.: A Multi-stream Adaptation Framework for Bandwidth Management in 3D Tele-immersion, *Proc. ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV '06)* (2006).
- 6) Hu, S.Y. and Liao, G.M.: Scalable peer-to-peer networked virtual environment, *Proc. ACM 3rd Workshop on Network and System Support for Games (NETGAMES '04)*, pp.129–133 (2004).
- 7) Vleeschauwer, B.D., Bossche, B.V.D., Verdickt, T., Turck, F.D., Dhoedt, B. and Demeester, P.: Dynamic Microcell Assignment for Massively Multiplayer Online Gaming, *Proc. 4th ACM Workshop on Network and System Support for Games (NETGAMES '05)* (2005).
- 8) Yu, A.P. and Vuong, S.T.: MOPAR: A Mobile Peer-to-Peer Overlay Architecture for Interest Management of Massively Multiplayer Online Games, *Proc. ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV '05)* (2005).
- 9) Eraslan, M., Georganas, N.D., Gallardo, J.R. and Makrakis, D.: A Scalable Network Architecture for Distributed Virtual Environments with Dynamic QoS over IPv6, *Proc. IEEE Symposium on Computers and Communications (ISCC 2003)* (2003).
- 10) Chertov, R. and Fahmy, S.: Optimistic Load Balancing in a Distributed Virtual Environment, *Proc. ACM Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '06)* (2006).
- 11) Yamamoto, S., Murata, Y., Shibata, N., Yasumoto, K. and Ito, M.: QoS Adaptation for Realizing Interaction between Virtual and real Worlds in Pervasive Network Environment, *Proc. ACM Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '07)* (2007).
- 12) Rowstron, A. and Druschel, P.: Pastry: Scalable, distributed object location and

- routing for large-scale peer-to-peer systems, *Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pp.329-350 (2001).
- 13) Yang, Z., Cui, Y., Yu, B., Liang, J., Nahrstedt, K., Jung, S. and Bajscy, R.: TEEVE: The Next Generation Architecture for Tele-Immersive Environments, *Proc. 7th IEEE International Symposium on Multimedia (ISM '05)* (2005).
 - 14) Yasumoto, K. and Nahrstedt, K.: Realistic Voice Chat Framework for Cooperative Virtual Spaces, *Proc. IEEE 2005 International Conference on Multimedia and Expo (ICME 2005)*, CD-ROM (2005).
 - 15) Kitasuka, T., Nakanishi, T. and Fukuda, A.: Wireless LAN based Indoor Positioning System WiPS and Its Simulation, *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM '03)*, pp.272-275 (2003).
 - 16) Scott, J. and Dragovic, B.: Audio Location: Accurate Low-Cost Location Sensing, *Proc. 3rd International Conference on Pervasive Computing (Pervasive 2005)*, LNCS 3468 (2005).
 - 17) Intel Research: Place Lab. <http://www.placelab.org/>
 - 18) Kouroggi, M. and Kurata, T.: A method of personal positioning based on sensor data fusion of wearable camera and self-contained sensors, *Proc. IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2003)*, pp.287-292 (2003).
 - 19) Yamamoto, S., Murata, Y., Yasumoto, K. and Ito, M.: A Distributed Event Delivery Method with Load Balancing for MMORPG, *Proc. 4th ACM Workshop on Network and System Support for Games (NETGAMES '05)* (2005).
 - 20) Tanenbaum, A.S. and Steen, M.V.: *Distributed Systems - Principles and Paradigms*, Prentice Hall (2002).

(平成 20 年 5 月 19 日受付)

(平成 20 年 11 月 5 日採録)



山本 眞也 (学生会員)

2006 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了。現在、同研究科博士後期課程在学中。P2P, 分散仮想環境の研究に従事。



村田 佳洋 (正会員)

1975 年生。2003 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年奈良先端科学技術大学院大学情報科学研究科助手。現在、広島市立大学情報科学研究科准教授。遺伝的アルゴリズム, エージェント技術等の研究に従事。



柴田 直樹 (正会員)

1996 年, 1998 年, 2001 年にそれぞれ大阪大学基礎工学部中退, 同大学院基礎工学研究科博士前期課程修了, 同大学院基礎工学研究科博士後期課程修了。2001 年より奈良先端科学技術大学院大学情報科学研究科助手。2004 年 1 月より滋賀大学経済学部情報管理学科講師。2004 年 4 月より現在, 滋賀大学経済学部情報管理学科助教授。分散システム, ITS, 遺伝的アルゴリズム等の研究に従事。ACM, IEEE 各会員。



安本 慶一 (正会員)

1991 年大阪大学基礎工学部情報工学科卒業。1995 年同大学院博士後期課程退学後, 滋賀大学経済学部助手。2002 年より現在, 奈良先端科学技術大学院大学情報科学研究科准教授。博士(工学)。分散システム, マルチメディア通信システムに関する研究に従事。ACM, IEEE/CS 各会員。



伊藤 実 (正会員)

1977 年大阪大学基礎工学部卒業。1979 年同大学院基礎工学研究科博士前期課程修了。1979 年より大阪大学基礎工学部助手。1986 年より大阪大学基礎工学部講師。1989 年より大阪大学基礎工学部助教授。1993 年より奈良先端科学技術大学院大学情報科学研究科教授。現在に至る。工学博士。関係データベース, オブジェクト指向データベースの理論等の研究に従事。ACM, IEEE, 電子情報通信学会各会員。