We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

QoS Adaptation for Realizing Interaction Between Virtual and Real Worlds Through Wireless LAN

Shinya Yamamoto[1], Naoki Shibata[2], Keiichi Yasumoto[3] and Minoru Ito[3]
[1]Tokyo University of Science Yamaguchi,
[2]Shiga University,
[3]Nara Institute of Science and Technology
Japan

1. introduction

Recently, there are many studies regarding to MR (Mixed Reality) and AR (Augmented Reality)(ARtoolkit; Fudono et al. (2005); Ichikari et al. (2006)). Many research efforts have also been made for NVE (Networked Virtual Environment) and CSCW (Computer Supported Cooperative Work) (Fujimoto & Ishibashi (2005); Yang et al. (2010)). These technologies allow remote users to participate in social activities such as shopping, exhibition, sports, and game which are held in real space. However, VE requires high performance servers and high speed networks for supporting massively many users. AR and MR often require special devices (e.g. a motion capturing suit, an immersive display, a camera array, and so on) for capturing movements of users in a real world and reflecting them into a virtual space. In order to utilize VE and AR/MR for various purposes, a technology for realizing these techniques with low cost is essential. In our target application systems, the data amount exchanged between users is much bigger than that in typical network games, because we would like to capture and reflect more realistic and complex movement of objects for smooth interaction. However, we should notice that the available communication bandwidth in the Internet is limited. We introduce a DVE (Distributed Virtual Environment) mechanism to solve these problems. In order to realize the DVE mechanism, the following five criteria should be satisfied: (1) real and virtual users share a common virtual space; (2) users can freely change their positions and directions, and the changes are instantly reflected in other users' views; (3) each user can introduce objects into the shared space, and make actions, such as pushing and holding objects, and reaction should be reflected in other users' views; (4) the required apparatuses should not be special nor expensive; and (5) a massive number of objects can exist in the shared space.

There are some studies on communication architectures for MMOG (Hu & Liao (2004); Vleeschauwer et al. (2005); Yu & Vuong (2005)) and NVEs for remote cooperation (Chertov & Fahmy (2006); Eraslan et al. (2003)). Some of these existing studies realize scalability on sharing virtual space between many users using P2P technologies. However, they allow sharing virtual space and objects among only virtual users. On the other hand, the existing MR and AR technologies, the criteria (1) to (3) can be satisfied. However, they require special devices, servers and networks, and thus satisfying the criteria (4) and (5) is difficult.

In this chapter, we propose a framework named FAIRVIEW (Yamamoto et al. (2007)) which realizes smooth cooperation and interaction between real and virtual users satisfying all the criteria (1) to (5) using inexpensive devices off the shelf. To satisfy the criteria (1) to (3), FAIRVEIEW produces a hybrid shared space by overlapping a virtual space and a real space, and provides a mechanism for allowing the virtual and real users to observe each other. To satisfy the criterion (4), we suppose that virtual users use ordinary PCs with an internet connection, and that real users use wearable computers with HMDs (head mount display) or smartphones, with internet connection via wireless LAN. In FAIRVIEW, the information regarding to orientations and positions of real objects (called *AR information*, hereafter) are measured at short intervals using an existing AR measurement tool. The information is exchanged among user terminals, and objects are displayed as 3D graphics on the display of the virtual user's terminal. To achieve the criterion (5), we propose a mechanism for delivering AR information as well as action/reaction to the objects in real time (which we call *AR event delivery mechanism*, hereafter) to realize smooth cooperation among real and virtual users. Here, information of action/reaction consists of information of attribute (the color or the form, etc), action type(push, pull, etc), action strength and action direction. In addition, we define the *AR event* that consists of AR information and the information of action/reaction. The AR event delivery mechanism includes a QoS adaptation mechanism for controlling the intervals of transferring AR events between users so that the total transmission rates will not exceed the limit of available bandwidth. The adaptation mechanism decides the transfer intervals according to importance of each object for each user, which is determined automatically according to the distance and position of the object in the user's view.

To evaluate the proposed mechanism, we analyzed the required bandwidths and investigated the effectiveness of our QoS adaptation mechanism under some configurations with different numbers of objects. In addition, in the real space, users may not get sufficient efficacy of QoS adaptation because users share wireless bandwidth in a single AP coverage area. However, this problem can be improved by introducing multiple APs with appropriate AP selection. Therefore, we conducted computer simulation for measuring improvement of QoS adaptation with strategic AP selection when there are multiple APs in the real space. Furthermore, in the real-time interaction, users often do the sudden action, such as turn around. Therefore, to evaluate tolerance of our QoS adaptation mechanism to the sudden action of users, we measured the delay until reflecting update frequency of objects when the user's view changes, by using a prototype of FAIRVIEW. Simultaneously, to evaluate whether FAIRVIEW can achieve sufficiently short delays for real-time interaction, we measured the end-to-end AR event delivery delays through the AR event delivery mechanism in LAN. As a result, we confirmed that the proposed mechanism realizes smooth interaction involving a large number of real and virtual users/objects on an ordinary wireless LAN and internet environment.

2. Related work

Studies regarding to NVE and DVE, can be roughly categorized into two types: one is aiming at reducing traffic of game information and keeping scalability by efficiently managing Area of Interest (AOI) of users, and another is aiming at efficiently using computation and network resources.

In Hu & Liao (2004); Vleeschauwer et al. (2005); Yu & Vuong (2005), efficient AOI (Area of Interest) management methods are proposed. In Hu & Liao (2004), game space is dynamically divided based on Voronoi diagram for direct communication among players in a same

fragment. In Vleeschauwer et al. (2005), the space is divided into small areas called micro cells. To distribute the load for processing events among multiple servers, the regions managed by each server are dynamically changed. In Yu & Vuong (2005), the shared virtual space is divided into honeycomb regions, and a mechanism based on Pastry (Rowstron & Druschel (2001)) is used to allow each user to receive information for players and objects in the player's AOI.

Although these existing NVEs enable efficient information exchange among virtual users, it is difficult to apply them to the mixed space of real and virtual worlds which requires a large amount of information exchange in real time on resource-limited wireless network, which is the target environment of FAIRVIEW.

In Chertov & Fahmy (2006); Eraslan et al. (2003), a load distribution and a QoS adaptation mechanisms for DVEs are proposed, respectively. In Chertov & Fahmy (2006), in order to cope with so called *area boundary problem* (inconsistency caused by neighboring areas managed by different servers), the whole shared space is managed by each of game servers, and game processing tasks are flexibly distributed among the servers. Unfortunately, this method supposes high performance servers and high-speed networks and treats only virtual users. In Eraslan et al. (2003), an IPv6-based network architecture called VESIR-6 is proposed for realizing a large-scale DVE where users can share a 3D virtual space and objects. Aiming at efficient utilization of network resources, VESIR-6 uses multicast for delivering object state updates to users, anycast for load distribution among servers, and IntServ/DiffServ-based QoS adaptation mechanism for regulating per-flow transmission rate. However, VESIR-6 does not suppose wireless network environment which is necessary for interaction between real and virtual users. Also, delivery of object state updates is managed only by joining/leaving the corresponding multicast group. Therefore, VESIR-6 cannot provide fine-grain QoS adaptation like the proposed method.

The most related study to our work is tele-immersion which captures the whole environment and reproduces it at geographically distant location. TEEVE (Yang et al. (2010)) displays 3D live visuals from each user's view in real-time, and constructs an environment for cooperative working. However, expensive and specialized devices and infrastructures such as 3D cameras and broadband network environments are needed to construct tele-immersion environment.

3. Overview of FAIRVIEW

This section overviews the functions of FAIRVIEW and presents example applications, then describes the target environment and the basic ideas for implementation.

3.1 Functions of FAIRVIEW

FAIRVIEW overlaps real space and virtual spaces, and provides the environment where users in real space (called *real users*) and users in virtual spaces (called *virtual users*) can interact as if they were in the same space. We call the overlapped space *hybrid space* [1] in Fig. 1. The main functions of FAIRVIEW are as follows: *Function (1) Providing the same view to real/virtual users:* Virtual users can change their positions in virtual space using keyboards and mice

---

[1] FAIRVIEW is also capable of overlapping more than one distant real spaces and provides the same view for the users in those spaces. For simplicity, we focus on interaction between the users in real space and virtual space.
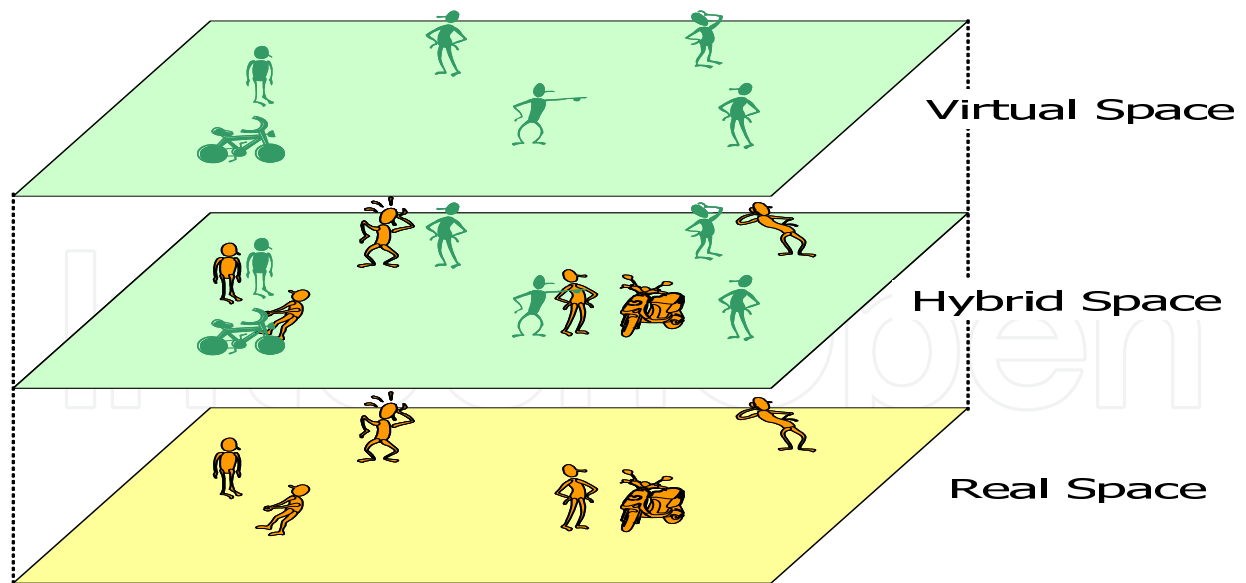
Fig. 1. Hybrid Space Produced by FAIRVIEW

like ordinary 3D first person shooting games. Real users can ordinarily move using their feet in real space. The users see objects in both virtual and real spaces according to their positions/directions. *Function (2) Voice conversation among users:* Users can talk with each other using their voice. The user's voice can be heard according to the user's position. This function can be realized with the technique in Yasumoto & Nahrstedt (2005). *Function (3) Object sharing:* Real objects and virtual objects can be registered in (and also unregistered from) the hybrid space. The registered objects can be seen by both virtual and real users. Users themselves are also objects[2]. Shared objects can be seen by all users whose views include the objects. We assume that 3D geometry data for the registered objects are prepared beforehand. *Function (4) Moving and bonding objects:* Users can move neighboring objects. Virtual objects can be moved by both real and virtual users, while real objects can be moved only by real users. Movement of objects can be observed by other users. Two or more objects can be bonded by specifying their relative positions. Bonded objects basically move together, but if a user moves a virtual part of bonded objects made of real and virtual objects, the bonded object is disengaged.

3.2 Applications of FAIRVIEW

We describe two example applications which are enriched by using the functions of FAIRVIEW described above.

Virtual flea market In this application, real and virtual users trade goods as if they were in a real market. Virtual users go shopping to a flea market in hybrid space. Real sellers register their goods to hybrid space. Virtual buyers see what kinds of goods are sold from distance. The sellers and buyers interact with each other via gestures and voice. The sellers show and explain details of their goods to virtual users by rotating and moving the goods. This kind of application includes exhibition, trade fair, and shopping center.

---

[2] Unregistered real objects can also be seen by real users, but in this chapter we assume that all objects are registered.

Multiplayer game In this application, virtual users participate in a paintball wargame held in real space. In paintball wargames, players possess airguns and targets, and shoot opponent groups' targets. Players whose targets are shot lose the game. Real players only need to register airgun and target. The virtual users cooperate with other real or virtual users. Virtual characters like a huge dinosaur can participate in the game. This kind of application includes attractions in theme parks, events on a street corner.

3.3 Devices and network used in FAIRVIEW

Table 1 shows the list of necessary equipments for the users. Real users use wireless small computing devices such as smartphones, and HMD with which the real view can be seen as the background of virtual view. These devices should not hinder real users' movements. The equipments also include positioning devices such as GPS receiver, sensors to detect position and direction, and audio input/output devices. We assume that inexpensive devices are available for these purposes. We use webcams and dedicated software like ARtoolkit (ARtoolkit) for positioning and detecting orientations of objects. Virtual users use ordinary PCs for FAIRVIEW.

Type	Computer	Display	Network	Other
Real User	PC/Smartphone	HMD/etc	WiFi/etc	Sensor, Webcam
Virtual User	PC	LCD/etc	Internet	Mouse/etc

Table 1. User Equipment

3.4 Basic ideas to implement FAIRVIEW

To realize an application using AR technology, we considered how the views seen by users are rendered inexpensively. In TEEVE (Yang et al. (2010)), images captured by 3D multi-cameras are processed and transferred through Internet2. Since we are aiming at inexpensively implementing the functions, we decided to measure AR information using inexpensive sensors. To realize this method, we have to resolve the following three problems: (i) measuring AR information accurately; (ii) transferring the measured AR information in real time; and (iii) rendering the objects.

For resolving the problem (i), we use GPS receiver if a user is in outdoor space. If a user is in indoor space, we use an indoor positioning method such as a method using wireless LAN access points (Kitasuka et al. (2003)), a method using speakers and microphones (Scott & Dragovic (2005)), Place Lab (Lab.), and Weavy (Kourogi & Kurata (2003)). The orientation of an object can be measured using rotational and translational acceleration sensors, methods based on image processing such as ARtoolkit (ARtoolkit) or the method in Fudono et al. (2005). To use the methods based on image processing, stickers with special patterns printed are put on objects, and the orientation and position of the objects can be obtained instantaneously from an image captured by a web camera. This kind of methods have cost advantage. In FAIRVIEW, AR information can be measured by combining the methods described above.

For the problem (iii), we assume that users have terminals capable of rendering 3D graphics. The problem (ii) is the main problem which we treat in this chapter. To resolve this problem, we need a delivery mechanism with which AR information can be exchanged among wireless

and wired users in real time. We call this mechanism *AR event delivery mechanism*. In applications such as flea market or paintball wargame, hybrid space may have many objects and bandwidth shortage can occur as the amount of exchanged AR information becomes large. Therefore, we have to lower the frequency of delivering AR information. However, if the frequency becomes too low, the user experience of interaction may be ruined. Therefore, we propose a QoS adaptation mechanism which maximizes users' satisfaction by prioritizing transfer of AR information among objects based on relative position and orientation between users and objects. As a solution, we present the AR event delivery mechanism in Sect. 4, and the QoS adaptation mechanism in Sect. 5.

4. AR event delivery mechanism

	notation	definition
Real Space	R	
Virtual Space	V	
Hybrid Space	H	$H = (R, V)$
Real Objects	RO	$RO = \{ro_1, ..., ro_n\}$
Virtual Objects	VO	$VO = \{vo_1, ..., vo_m\}$
Objects	o	$o \in RO \cup VO$
Real Users	RU	$RU = \{ru_1, ..., ru_l\}$
Virtual Users	VU	$VU = \{vu_1, ..., vu_k\}$
Users	u	$u \in RU \cup VU$
User Terminals	$node(u)$	
Real User Terminals	RN	$RN = \{node(u) \| u \in RU\}$
Virtual User Terminals	VN	$VN = \{node(u) \| u \in VU\}$
All Object	AO	$AO = RO \cup VO \cup RU \cup VU$
Object including Users	ao	$ao \in AO$
AR Information	$AR(ao)$	$AR(ao) = \{pos, angle\}$
Attribute Information	$attribute$	$attribute = \langle color, form, ...\rangle$
Action	$action$	$action = \langle type, direction, strength, ...\rangle,$ $type \in \{push, pull, ...\}$
AR Events	$ARe(ao)$	$ARe(ao) = (AR(ao), attribute, action)$

Table 2. Definitions

This section describes details of the AR event delivery mechanism.

4.1 Notation

Let R and V be the target real space and the corresponding virtual space, respectively. Let $H = (R, V)$ be the hybrid space produced by overlapping R and V. We suppose that H is an axis-aligned rectangle on x-y plane of 3D coordinate system. Let $hvec = (1, 0, 0)$ and $vvec = (0, 0, 1)$ be the horizontal standard vector and the vertical standard vector of H, respectively.

Let $RO = \{ro_1, ..., ro_n\}$, $VO = \{vo_1, ..., vo_m\}$, $RU = \{ru_1, ..., ru_l\}$ and $VU = \{vu_1, ..., vu_k\}$ be the set of real objects in R, the set of virtual objects in V, the set of real users (i.e., mobile users) and the set of virtual users (i.e., remote PC users), respectively. In addition, let AO be the set of all objects including users, where each member object is denoted by $ao \in AO$.
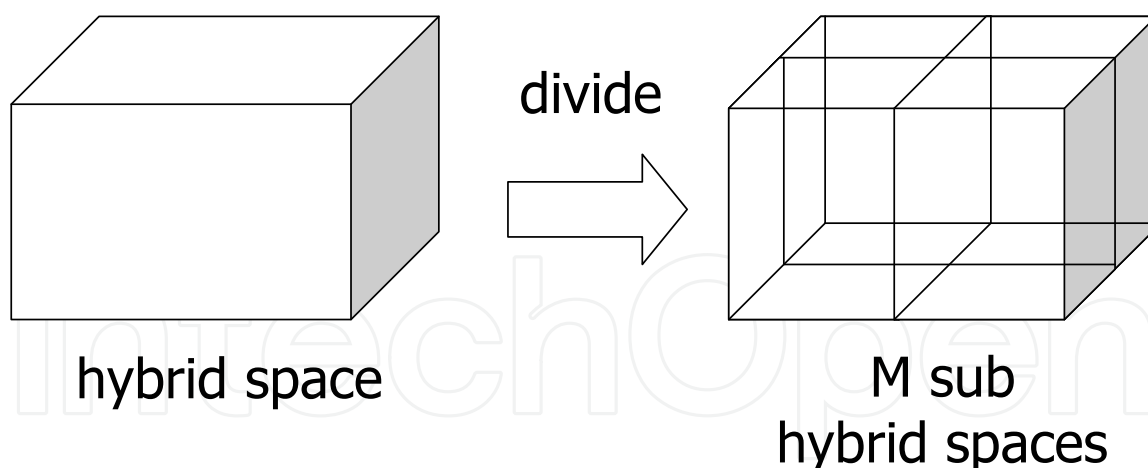
Fig. 2. Division of Shared Space

Note that $RU \subseteq RO$ and $VU \subseteq VO$. Let $node(u)$ be the user terminal of u for each user $u \in RU \cup VU$. $RN = \{node(u)|u \in RU\}$ and $VN = \{node(u)|u \in VU\}$ are the real user terminals and the virtual user terminals, respectively.

For each object $ao \in AO$, let $AR(ao) = (pos, angleH, angleV)$ be the AR information of ao, where pos, $angleH$, and $angleV$ are its position on or beyond H, the horizontal angle to $hvec$, and the vertical angle to $vvec$, respectively. Each item of AR information is referred to by, e.g., $ao.pos$. In addition, let $ARe(ao) = (AR(ao), attribute, action)$ be the AE Event of ao, where $AR(ao)$ is the AR information of ao, $attribute = \langle color, form, ... \rangle$ is attribute information such as color and so forth, and $action = \langle type, direction, strength, ... \rangle$ is action to other objects, characterized by $type \in \{push, pull, ...\}$.

We suppose that each real user terminal in RN can measure its user's AR information at a sufficiently high frame rate, e.g., 60 times per second, with equipment explained in Sect. 3. We also suppose that AR information of each real object ro except for users can be measured by a user terminal in ro's proximity with ARtoolkit and webcam.

4.2 Assumption on user communication

We suppose that the whole real space R is covered by only one AP connected to the Internet. Therefore, each real user terminal in RN can communicate with any virtual user terminal in VN.

Let BW_{AP} be the available bandwidth between real user terminals and AP. Note that all real user terminals share the bandwidth. For each $vn \in VN$, let $bw_{AP}(vn)$ be the available bandwidth between vn and AP. For each $vn \in VN$ and $rn \in RN$, let $bw(vn, rn)$ be the available bandwidth between vn and rn. Note that $bw(vn, rn) = Min(BW_{AP}, bw_{AP}(vn))$. For each pair of virtual user nodes $vn1, vn2 \in VN$, let $bw(vn1, vn2)$ be the available bandwidth between $vn1$ and $vn2$.

4.3 AR event delivery mechanism

We choose some of the user terminals as server nodes to manage efficient AR events exchange. In order to reduce processing and traffic load of each server node based on user's AOI, we
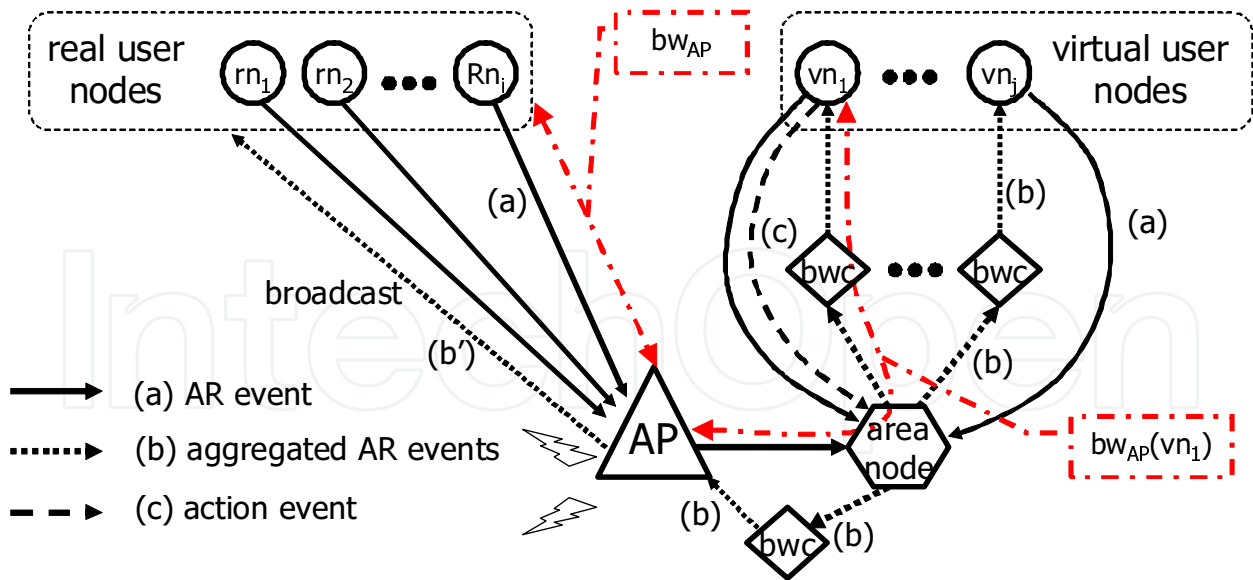
Fig. 3. Overlay Network for AR Event Delivery

divide the whole shared space H into small rectangular sub-areas, as shown in Fig. 2, and assign a server node called *area node* to each sub-area. This is a similar approach to existing P2P-based MMOG gaming architectures such as Yamamoto et al. (2005). Let an_A be the area node assigned to sub-area A.

The sub-area an_A receives AR events of objects in A, and delivers the AR events to users (in A and neighboring sub-areas) who are watching the objects.

In FAIRVIEW, since the network resource is tight due to wireless communication and calculating the reaction of an object as a consequence of an action is heavy for a real user terminal, we introduce new server node called *bandwidth controller node (bwc-node)*.

One bwc-node is prepared for each virtual user terminal $vn \in VN$ satisfying that $bw(vn, an_A)$ is less than necessary traffic for AR event transmission or for the set of all real user terminals[3]. Let $bn(u)$ be the bwc-node for user $u \in RU \cup VU$. Then $bn(u)$ applies QoS adaptation to the AR event stream between an_A and $node(u)$, by monitoring $bw(bn(u), node(u))$.

Consequently, we construct the overlay network per sub-area, consisting of an area node, N object nodes, M user nodes, and M bwc-nodes, as shown in Fig. 3, where N and M are the numbers of objects and users in the subarea, respectively. We will explain how these nodes exchange AR events below.

User Node   In FAIRVIEW, (1) a user $u \in RU \cup VU$ can watch other objects in its view, (2) u can be watched by other users since u is also an object, and (3) u can take an action to other objects.

For (1), $node(u)$ measures u's AR information continuously and if the information differs from the last measurement, $node(u)$ sends the information as AR event to the area node an_A (Fig. 3 (a)). For (2), $node(u)$ receives the AR events of the objects in u's view and draws

---

[3] In FAIRVIEW, due to wireless bandwidth limitation, we restrict all real users to watching each virtual object at the same framerate.

the latest appearance of objects on display of $node(u)$. To receive AR event, $node(u)$ uses publish/subscribe model (Tanenbaum & Steen (2002)). Once $node(u)$ sends its AR event to the area node an_A, an_A automatically identifies the objects in u's view, and forwards the AR events of the objects to $node(u)$ via the bwc-node (Fig. 3 (b)(b')). As shown in Fig. 3 (b'), the real user terminals receive the AR events from the bwc-node assigned for real space R via wireless AP by broadcast. For (3), when u takes an action to virtual object o, $node(u)$ sends an action event containing power and direction, to an_A (Fig. 3 (c)).

Area Node For each sub-area A of the hybrid space H, a virtual user node is selected and assigned to the area node. A virtual node with sufficient network and computation resources is selected, e.g., by the lobby server when the application starts [4].

The area node an_A manages the positions of the objects as well as the users' view in the sub-area A. When an_A receives the AR events of all objects in A from the corresponding user nodes and object nodes (Fig. 3 (c)), an_A identifies the users who are watching part of other sub-areas neighboring A, based on their views calculated from their AR events (e.g., *pos* and *angleH*), and sends the AR events to the neighboring area nodes if needed. The sub-area an_A also receives the AR events of such users in other sub-areas from the neighboring area nodes. Finally, an_A sends the AR events of the objects in A to the user nodes via the corresponding bwc-nodes.

Bandwidth Controller Node One bwc-node is assigned to each user node u, although one node may serve as the bwc-nodes of multiple users. The bwc-node $bn(u)$ monitors available bandwidth to its associated user node $node(u)$, and regulates transmission rates of AR event streams.

4.4 The overall flow

The overall flow is as follows.

Step(1) User Node $node(u)$ sends AR event $ARe(u)$ to Area Node an. Then, when User Node $node(u)$ has the managerial responsibility of an object o, it acquires the AR information of o, makes AR event $ARe(o)$, and sends $ARe(o)$ to an.

Step(2) Area Node an adds an importance to AR event $ARe(ao)$ using the QoS adaptation mechanism described in Sect. 5.

Step(3) Area Node an sends to Bandwidth Control Node bn, AR event $ARe(ao)$ with the importance.

Step(4) Bandwidth Control Node bn regularly redistributes bandwidth with the QoS adaptation mechanism explained in Sect. 5 and decides update frequency of each AR event $ARe(ao)$.

Step(5) Bandwidth Control Node bn sends the set of AR events to User Node $node(u)$ with the update frequency defined in Step(4).

Step(6) User Node $node(u)$ updates data of Hybrid Space H with the set of received AR events.

Step(7) User Node $node(u)$ draws Hybrid Space H with the updated data.

Step(8) returns to Step(1).

---

[4] The node selection/replacement algorithm for area nodes and bwc-nodes are omitted due to the space limitation. We use a similar technique to Yamamoto et al. (2005).
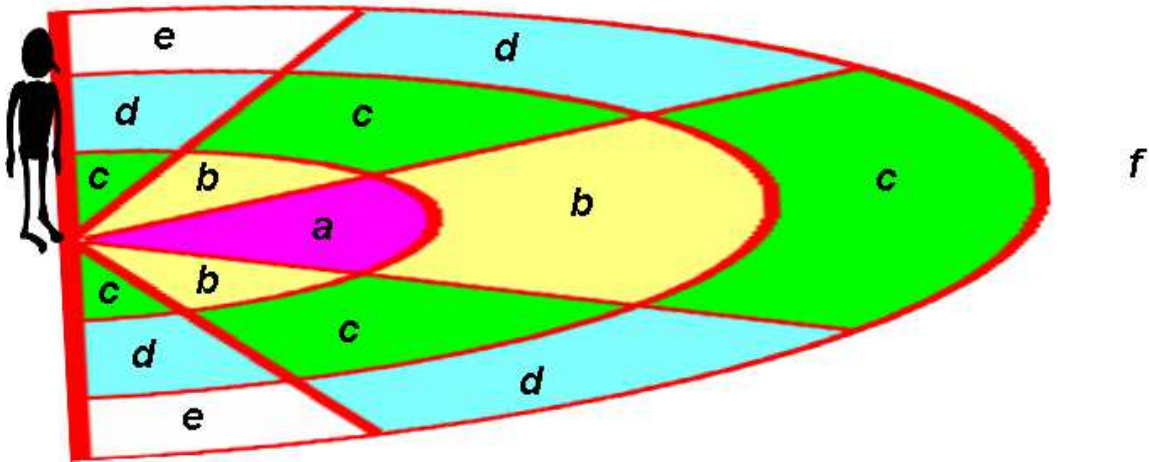
Fig. 4. User's View and Zones with Relative Importance

5. View-oriented QoS adaptation

The basic ideas of our QoS adaptation mechanism are as follows: (1) we decide relative *importance value* of each object according to how important the object is for a user; and (2) for each user, we regulate transmission rates of AR events of observable objects based on their importance values so that the sum of transmission rates are less than the available bandwidth.

5.1 Decision of importance value

Let $Watcher(o)$ be the set of users who can observe an object o in $RO \cup VO$. The set of users $Watcher(o)$ is defined by

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

Where $View(u)$ is u's view on hybrid space H, and represented by a half circle as shown in Fig. 4.

For each object o in $RO \cup VO$ and each user u in $Watcher(o)$, let $Imp(o, u)$ be the importance value of o for u. Like in the real world, the importance value $Imp(o, u)$ should increase as the distance between o and u is shorter and o is located nearer the center of u's view. Therefore, we define each user's view as a half circle and divide it into 15 zones with five levels of importance a, b, c, d and e, as shown in Fig. 4. Here, the objects located on zone a have the highest importance, and the importance of objects on zones b, c, d and e decreases in this order.

5.2 Decision of framerates of AR events

The transmission rate of AR events for each object towards user u is decided based on the ratio of its importance value to the sum of importance values of all objects in u's view. The available bandwidth is distributed among the objects, and the framerate of AR events for each object is decided from the assigned transmission rate and the size of each AR event.

As mentioned in Sect. 4, AR events of each object are delivered to each user node through the corresponding bwc-node (see Fig. 3). The bwc-node drops packets of the AR events so that the transmission rate does not exceed the assigned bandwidth.

We describe our view-oriented QoS adaptation mechanism using examples. The QoS adaptation for AR event streams differs depending on the receiver type (i.e., virtual user or real user). Thus, we give two examples in the following subsections.

5.2.1 QoS adaptation for virtual user

Suppose that a virtual user $v \in VU$ is watching three objects o_1, o_2 and o_3 in his view. In this case, $node(v)$ receives the AR events of those three objects via the bwc-node $bn_v \in VN$ (see Fig. 3).

We assume that the available bandwidth between bn_v and $node(v)$ is 1Mbps. We also assume that transmission rates for AR event streams of objects o_1, o_2 and o_3 are 0.5 Mbps respectively, that is, the sum of the streams is 1.5 Mbps. In this case, the available bandwidth (1 Mbps) is distributed according to the importance values of objects o_1, o_2 and o_3. Suppose that the importance values of objects o_1, o_2, and o_3 are 10, 25, and 15, respectively. As a result, portions of bandwidth $\frac{10 \times 1Mbps}{10+25+15} = 0.2Mbps$, $\frac{25 \times 1Mbps}{10+25+15} = 0.5Mbps$, and $\frac{15 \times 1Mbps}{10+25+15} = 0.3Mbps$ are assigned to the AR event streams, respectively. Based on this result, the bwc-node bn_v controls transmission of the AR event stream of each object by dropping some of the received packets.

5.2.2 QoS adaptation for real user

Suppose that a real user $r \in RU$ is watching three virtual objects o_1, o_2 and o_3 in his view. In this case, $node(r)$ receives the AR events of those three objects via wireless AP and the bwc-node $bn_W \in VN$ (see Fig. 3).

In this case, we define the importance value $Imp_r(o)$ of an virtual object $o \in VO$ as the sum of the importance rated by the users observing the object o, i.e., $Imp_r(o) = \sum_{u \in Watcher(o)}(Imp(o, u))$. The framerate of AR events of each object is decided similarly to the case in Sect. 5.2.1, and the bwc-node bn_W applies the QoS adaptation to the AR event streams to real users.

6. Experiments

In order to evaluate the proposed method, we conducted experiments on five items. First, we measured the required bandwidth for application of users and objects (Sect.6.2). Second, we measured the framerates at which users can watch the objects for two cases with and without our QoS adaptation mechanism (Sect.6.3). Thirdly, we measured the improvement of QoS adaptation with emulation in which users choose AP according to a strategy when there are multiple APs (Sect.6.4). Fourth, we measured the delay necessary for changing update frequency by QoS adaptation when the user's view changes, under some configurations with different numbers of objects using a prototype of FAIRVIEW (Sect.6.5). Finally, we measured the end-to-end delay through AR event delivery mechanism using a prototype of FAIRVIEW (Sect.6.5). Note that the experiments in Sect.6.2, 6.3 and 6.4 were emulations. The experiment in Sect.6.5 was conducted on a testbed using a prototype of FAIRVIEW with LAN.

6.1 Configurations

		Exp.6.2, 6.3	Exp.6.4	Exp.6.5
User	Real	100	100	–
	Virtual	100	–	1
Field Size (m^2)		50×50	50×50	6×6– 120×120
the number of APs		1	2, 4	–
Bandwidth ($Mbps$)	wired	10	–	–
	wireless	5	5	5
Virtual Objects		50– 5000	500– 2500	8– 1683
Packet Size		32	64	32– 64
Max update frequency ($frame/sec$)		60	60	30

Table 3. Experimental Configurations

We show the experimental configurations in Table 3. The sizes of virtual space V and real space R are both 50×50 m. As user's view in Fig. 4, we set the radius of a half circle to be 15m and divided the half circle equally so that the angle and the radius of each zone are 1/3 (i.e., $\pi/3$ and 5m), respectively. In this configuration, the average number of objects in each user's view is around 1– 2 when there are 50 objects in the whole space. In Fig. 4, zone c exceeds in the number of objects compared with the zones a, b because zone c is the largest in the gross floor area. Therefore, in zones a, b, there is not much advantage than in zone c when we give 2 times importance sequentially backward from zone e. Hence, we give zones a, b importance of 4 times sequentially from c backward. Thus, we assign the importance 64, 16, 4, 2, 1 to the zones a, b, c, d, e, respectively.

6.2 Required bandwidth for user terminals

The experimental configurations are shown in Table 3. The other setting is as follows. There is one wireless AP whose radio range covers the whole real space R. The available wireless bandwidth BW_{AP} is shared among all real users. One area node and bwc-nodes are allocated on PCs on fixed wired network. 100 real users, 100 virtual users, and n virtual objects are placed on the positions decided at random in the hybrid shared space, changing n from 50 to 5000. The directions of the users are also set randomly. Each user terminal sends 60 packets of AR events every second, and the size of each packet is 32 bytes. The result is shown in Fig. 5. This result is average of 100 trials.

We measured the required network bandwidth between a user node and the corresponding bwc-node under the configurations in Sect. 6.1 for both cases with and without our QoS adaptation mechanism. The average results of 100 simulations are shown in Fig. 5.

Without the QoS adaptation mechanism, the required bandwidths for each real user terminal ("RU without QoS" in Fig. 5) and each virtual user terminal ("VU without QoS" in Fig. 5) exceeded the capacity (i.e., $10Mbps$ and $5Mbps$) when the number of objects in the whole space becomes more than 250 and 500, respectively. With our QoS adaptation mechanism, the required bandwidth is regulated below the capacity even if the number of objects increases to 5000 ("RU with QoS" and "VU with QoS" in Fig. 5).
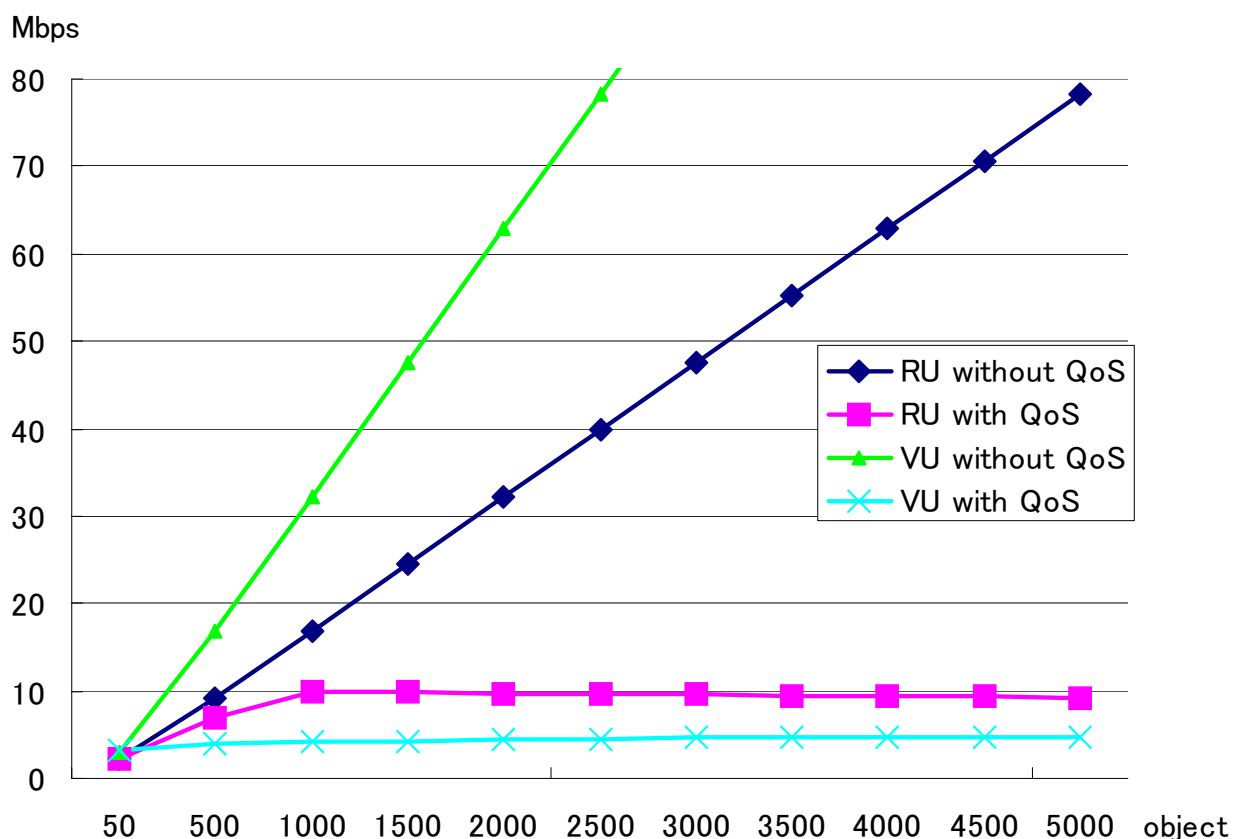
Fig. 5. Required Bandwidth for User Terminals

6.3 Impact of QoS adaptation

With our QoS adaptation method, AR events of more important objects are transmitted at higher transmission rates (i.e., framerates) than other objects, within the available bandwidth. To examine the effect of the QoS adaptation, we measured the framerates of AR events for objects in view zones a to e under the configurations in Sect. 6.2. The results are shown in Fig. 6 and Fig. 7.

In Fig. 6 and Fig. 7, the lines with labels a, b, c, d, and e show the average framerates for objects on the corresponding zones, and the line with label *uniform* shows the average framerate when the bandwidth is uniformly distributed to objects.

Fig. 6 shows that each virtual user can watch important objects in zones a and b at higher framerates than *uniform*. Especially, framerate of objects in zone a keeps more than 50 frames/sec while the number of objects is less than 4500. The framerates of less important objects on zones c, d and e are reduced below *uniform*.

Fig. 7 shows that each real user can also watch important objects in zones a at better framerates than *uniform*. However, the effect is smaller than the case for virtual users. The framerates of objects in other zones are reduced below *uniform*. This is because each object's importance value is decided as the maximum value among its watchers as explained in Sect. 5.1 and large portion of objects are regarded as important objects when the number of objects is large.
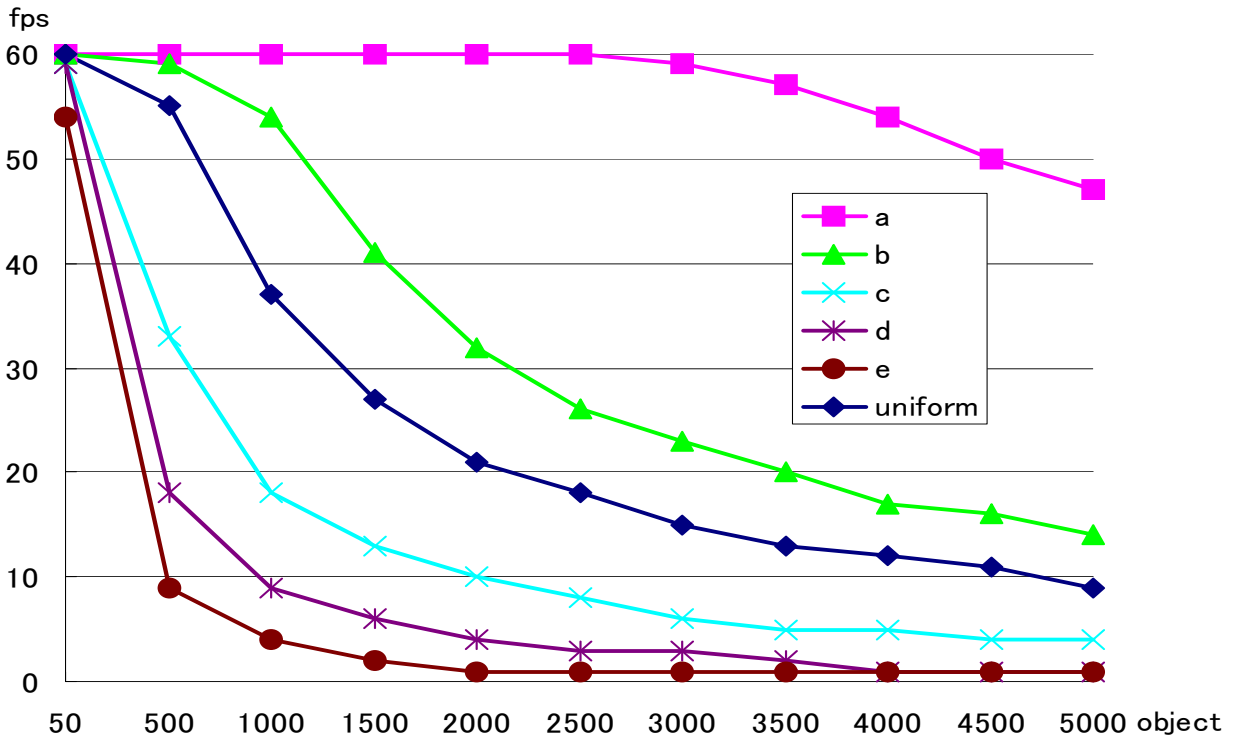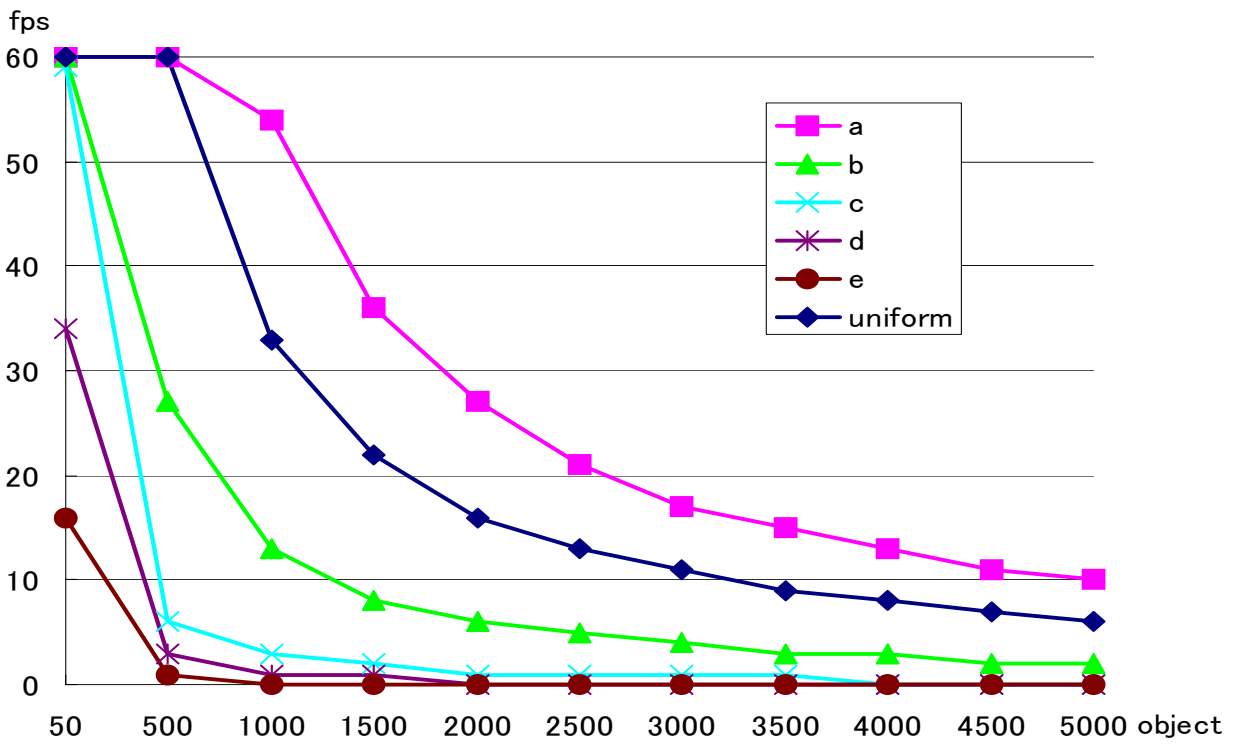
Fig. 6. Framerates Obtained by Virtual Users

Fig. 7. Framerates Obtained by Real Users

However, the framerates of the objects in zone *a* are improved to a great extent when the number of objects is less than 2000.

6.4 Improvement of QoS adaptation control by AP selection

The real users share AP and receive AR events broadcast from AP. In the real space, each object's overall importance value is the sum of the values rated by all users sharing AP. Therefore, it cannot sufficiently reflect the importance of each user (shown as Sect. 5.2.2 and Sect. 6.3). If multiple APs are installed in real space, and users can choose AP which can reflect one's importance effectively, then we can improve the satisfaction of users. Therefore, we measured the improvement of QoS adaptation with simulation in which users choose AP according to a strategy when there are multiple APs. In this experiment, we installed 2 to 4 APs. In case the number of AP is 2, we equally divide real space in x-axis into two areas and placed one AP in each area. In case the number of AP is 4, we placed AP in each of the two by two matrix regions formed by equally sectoring a long x-axis and y-axis. In this experiment, to focus on the AP selection strategy, we supposed that a radio range is the whole area. In this experiment, the strategies that users are supposed to use are as follows.

1. UNIFORM: the user connects to AP with ID equal to *his/her ID* mod *the number of APs*.
2. POSITION: the user connects to the nearest AP from his/her position.
3. VIEW: the user connects to the nearest AP from his/her position taking into account his/her view. Thus, comparing the his/her position and the end point of his/her view (e.g. distance = $15m$ and angle = $0°$), it employes the point whose distance to the AP is greater.

The results are shown in Fig. 8. This figure is typical boxplot that show how much frame-rate was assigned according to importance with strategic AP selection and the number of the AP. In this figure, the bar graph is Max-Min range and the box graph is the inter-quartile range. For example, in importance 64, 2 APs, "UNIFORM" in Fig. 8, we can see that there are the users for whom only 50 frames/sec was assigned to objects of importance 64, while many users can watch objects of importance 64 in 60 frames/sec. In this result, we can see how efficiently bandwidth allotment reflects the importance of the object for each user. In "UNIFORM" in Fig. 8, when several APs are available, the improvement is seen in comparison with the result in case the number of objects equals 2500 of Fig. 7. The value of framerates to be given disperse, because this strategy is random selection. In "POSITION" or "VIEW" in Fig. 8, dispersion of provided framerates for the importance of the user becomes small. In these results, "VIEW" was approximately equal to "POSITION". "VIEW" strategy removes users facing the other APs than the AP concerned. In spite of that, it is seen in Fig. 8, that bandwidth assignment in "VIEW" strategy seems more dispersive than "POSITION" strategy. The reason is that in "VIEW" strategy AP accepts far-off users looking towards itself as compensation does not accept far-off users looking outwards. As a result, the AP selection in consideration of the orientation of the user is effective in the QoS adaptation control for the real users. In QoS adaptation control for the real space, when there are some APs are available, we can get big benefits with strategic AP selection.

6.5 Processing delay of QoS adaptation

We measured the the coordination time of update frequency under some configurations when moving user's view by prototype system with different numbers of objects in LAN.
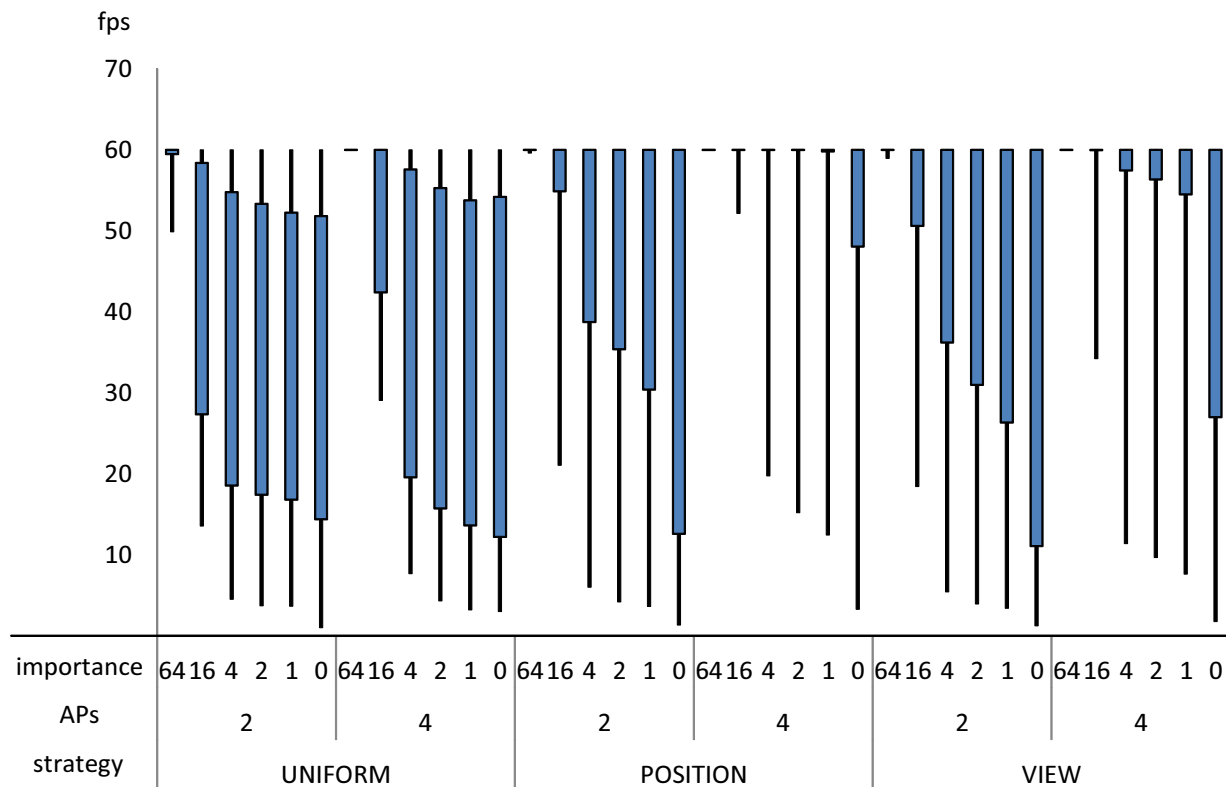
Fig. 8. Impact of AP Selection Strategy

Experimental configurations are shown in Table 3. The other setting is as follows. The user is placed in the center of the space. He/She walks at 1 m/sec towards the destination decided at random. When he/she arrives at the destination, the next destination is decided at random again. In this experiment, we place n virtual objects every 3 meters from the user. Then, the space has just only enough room for the all objects. Updates of the data of each object by the AR event, the transmission of packets, reception of packets, changes of the importance, and QoS adaptation control are preformed by parallel computation at about 30 times per second in the prototype for an experiment. Then, each process is asynchronous processed in an individual timeslot. The change of the importance is executed in a bwc-node and the QoS adaptation control is executed in an area node. We used TCP for communication protocol. The specifications of PCs are as follows: the area node and the bwc-node : CPU:Athlon64 X2 4200+, Memory:2GB, OS:Debian Linux (kernel 2.6.8); and a user node: CPU:Opteron 242(1.6 GHz)×2, Memory:8GB, OS:Debian Linux (kernel 2.6.8). Programming Language is JAVA SE 1.6.0_05.

We measured, using the timestamp, the processing delay necessary for the recalculation of the importance and the recomputation of QoS adaptation control. In addition, we measured the end-to-end delay by the difference of timestamp from the time when user node sends AR event to the time when user node receives and updates data. These data do not include the delay of thread generation. The results are shown in Figs. 9 and 10. These results are average of 5 trials.
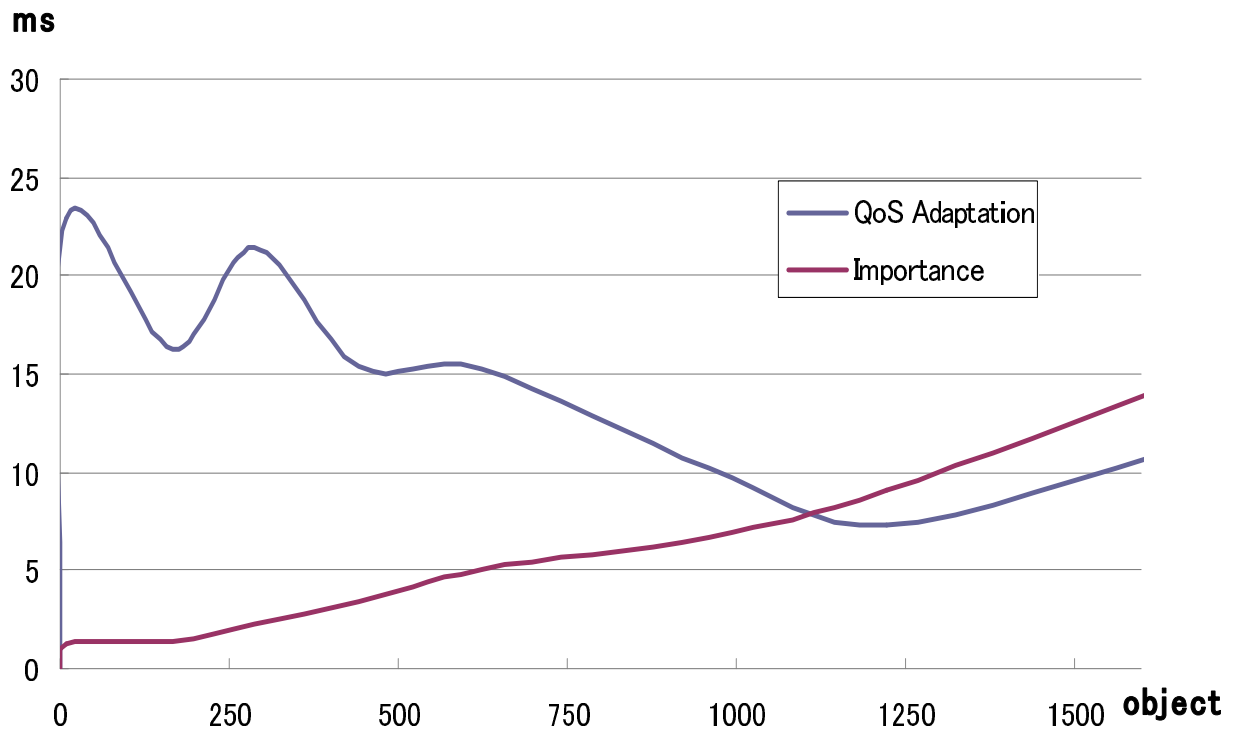
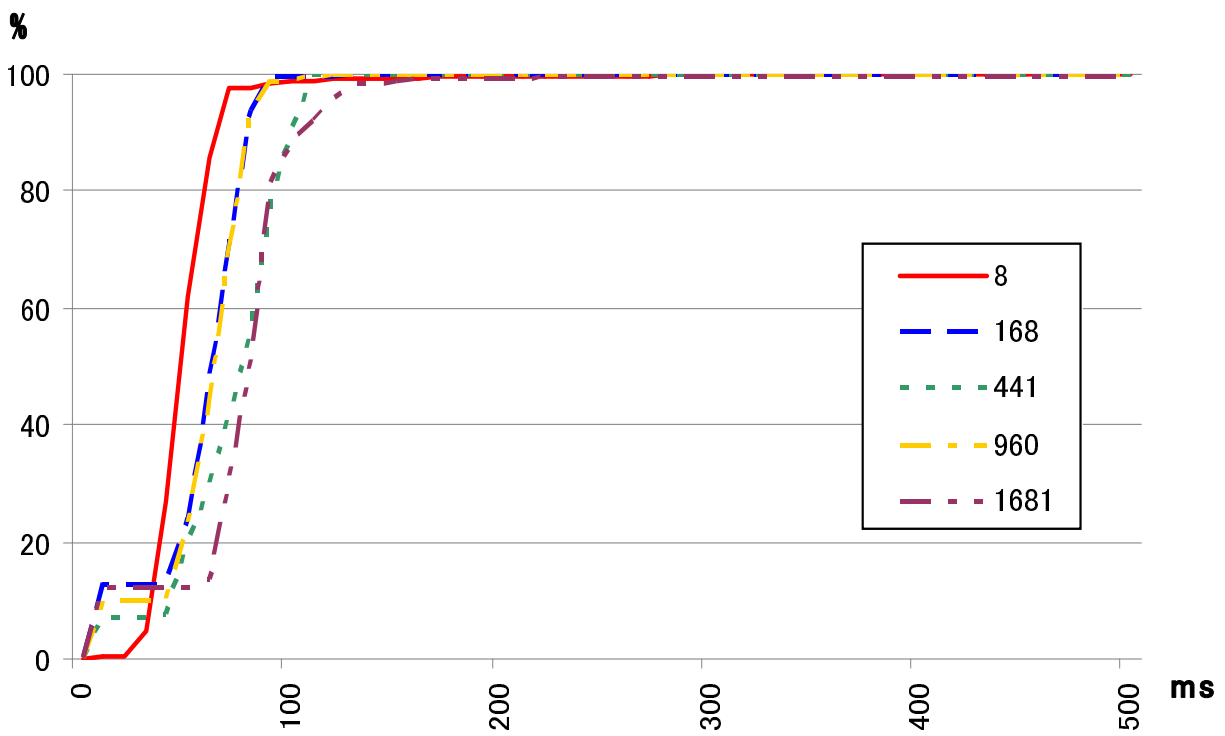Fig. 9. Delay by QoS Adaptation

Fig. 10. Cumulated Distribution of End - End Delay

Fig. 9 is a graph of the processing delay of recomputation of the importance to change along the position of the avatar ("importance" in Fig.9) and recomputation by QoS adaptation control ("QoS Adaptation" in Fig.9). As a result, in the recalculation of the importance, a processing delay considerably grows as the number of objects increases because computational complexity increases according to the number of objects. The main cause of this processing delay is calculation of distance and angle of the avatar vs. the object. Therefore, when there are many objects, the importance is difficult to calculate in short time (e.g., as for every frame). As a solution, we can increase the number of area nodes or lower the frequency of the recalculation. The delay of QoS adaptation control changes by a timing of the processing. However, this delay stay lower than 25ms and does not depend on the number of objects. Furthermore, by the total of the recalculation of the importance and recomputation of QoS adaptation control, we can see the tolerance of user's sudden action (such as turns around), how much delay the influence is reflected on QoS adaptation when the importance of each object largely changes.

In Fig. 10, we show the end-to-end delay through AR event delivery mechanism including a result of Fig. 9 (Step(1)– (7) in Sect. 4.4). The end-to-end delay is quite satisfactorily and less than 200ms. The delay can be factored as follows: the recalculation of importance, the recomputation of QoS adaptation control, the wait time of the asynchronous processing. In Fig. 9, we see that the sum of the recomputation of QoS adaptation control and recalculation of importance is less than 50ms. Therefore, we show a main cause is wait time of the asynchronous processing. In addition, in Figs. 9 and 10, when there are objects more than 1,500, the total calculation delay is about 30ms and the influence is reflected within 200ms. If a human turns around and considers movement to adjust a focus to an object , we can permit this delay in practical uses. As a result, our framework can sufficiently cope with games requiring quick response.

7. Conclusion

In this chapter, we proposed a framework for interaction between real and virtual users in hybrid shared space called FAIRVIEW. In FAIRVIEW, the users share the information about moving objects including users themselves with Publish/Subscribe method on divided sharing space. In addition, in order to use FAIRVIEW in the ordinary environments, we proposed a QoS adaptation mechanism that can be implemented in a network with bandwidth limitation.

Through experiments, we confirmed that our method can handle hundreds of real and virtual users and thousands of objects with sufficiently fast framerate in an ordinary wireless LAN and internet environment. In addition, we confirmed FAIRVIEW could effectively offer QoS adaptation for real users by introducing strategic AP selection among a number of APs, compared with the case in which QoS adaptation is made through a single AP.

The future work is as follows. In the wireless communication environment, the delay tends to greatly fluctuate in comparison with the Internet. To such environments, we need to devise methods for realizing a short latency. In addition, e.g. in game applications, targets and obstacles are treated equally because our QoS adaptation control method treats all objects based on their values of importance. We will examine adjustment of the importance according

to the kind of the object. Furthermore, we are planning to implement the proposed method as middleware library and evaluate its performance.

8. References

ARtoolkit . URL: *http://www.hitl.washington.edu/artoolkit/*

Chertov, R. & Fahmy, S. (2006). Optimistic load balancing in a distributed virtual environment, *ACM Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '06)* pp. 74– 79.

Eraslan, M., Georganas, N. D., Gallardo, J. R. & Makrakis, D. (2003). A scalable network architecture for distributed virtual environments with dynamic qos over ipv6, *IEEE Symposium on Computers and Communications (ISCC '03)* pp. 10– 15.

Fudono, K., Sato., T. & Yokoya, N. (2005). Interactive 3-d modeling system using a hand-held video camera, *14th Scandinavian Conference on Image Analysis (SCIA2005)* pp. 1248– 1258.

Fujimoto, M. & Ishibashi, Y. (2005). Packetization interval of haptic media in networked virtual environments, *4th ACM Workshop on Network and System Support for Games (NETGAMES '05)* .

Hu, S. Y. & Liao, G. M. (2004). Scalable peer-to-peer networked virtual environment, *ACM 3rd Workshop on Network and System Support for Games (NetGames '04)* pp. 129– 133.

Ichikari, R., Kawano, K., Kimura, A., Shibata, F. & Tamura, H. (2006). Mixed reality pre-visualization and camera-work authoring in filmmaking, *5th International Symposium on Mixed and Augmented Reality (ISMAR '06)* pp. 239– 240.

Intel Research Lab. URL: *http://www.placelab.org/*

Kitasuka, T., Nakanishi, T. & Fukuda, A. (2003). Wireless lan based indoor positioning system wips and its simulation, *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03)* pp. 272– 275.

Kourogi, M. & Kurata, T. (2003). A method of personal positioning based on sensor data fusion of wearable camera and self-contained sensors, *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2003)* pp. 287–292.

Rowstron, A. & Druschel, P. (2001). Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, *18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)* pp. 329–350.

Scott, J. & Dragovic, B. (2005). Audio location: Accurate low-cost location sensing, *3rd International Conference on Pervasive Computing(Pervasive 2005)* pp. 1–18.

Tanenbaum, A. S. & Steen, M. V. (2002). *Distributed Systems - Principles and Paradigms*, Prentice Hall.

Vleeschauwer, B. D., Bossche, B. V. D., Verdickt, T., Turck, F. D., Dhoedt, B. & Demeester, P. (2005). Dynamic microcell assignment for massively multiplayer online gaming, *4th ACM Workshop on Network and System Support for Games (NETGAMES '05)* .

Yamamoto, S., Murata, Y., Yasumoto, K. & Ito, M. (2005). A distributed event delivery method with load balancing for mmorpg, *4th ACM Workshop on Network and System Support for Games (NETGAMES '05)* .

Yamamoto, S., Shibata, N., Yasumoto, K. & Ito, M. (2007). Qos adaptation for realizing interaction between virtual and real worlds in pervasive network environment, *ACM*

Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '07)
pp. 119–124.

Yang, Z., Wu, W., Nahrstedt, K., Kurillo, G. & Bajcsy, R. (2010). Enabling multi-party
3d tele-immersive environments with viewcast, *ACM Transactions on Multimedia
Computing, Communications, and Applications (TOMCCAP)* 6(2): 1–30.

Yasumoto, K. & Nahrstedt, K. (2005). RAVITAS: Realistic voice chat framework for
cooperative virtual spaces, *IEEE 2005 International Conference on Multimedia and Expo
(ICME 2005)* pp. CD–ROM.

Yu, A. P. & Vuong, S. T. (2005). Mopar: A mobile peer-to-peer overlay architecture for interest
management of massively multiplayer online games, *ACM Network and Operating
System Support for Digital Audio and Video (NOSSDAV '05)* pp. 99–104.
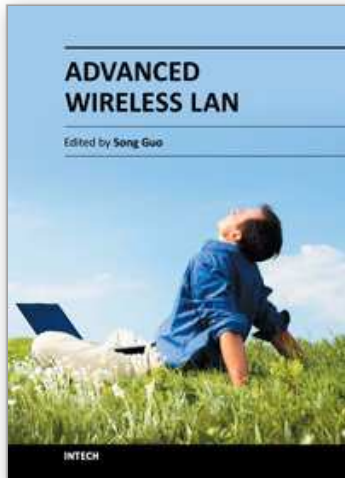
Advanced Wireless LAN

Edited by Dr. Song Guo

The past two decades have witnessed starling advances in wireless LAN technologies that were stimulated by its increasing popularity in the home due to ease of installation, and in commercial complexes offering wireless access to their customers. This book presents some of the latest development status of wireless LAN, covering the topics on physical layer, MAC layer, QoS and systems. It provides an opportunity for both practitioners and researchers to explore the problems that arise in the rapidly developed technologies in wireless LAN.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Shinya Yamamoto, Naoki Shibata, Keiichi Yasumoto and Minoru Ito (2012). QoS Adaptation for Realizing Interaction Between Virtual and Real Worlds Through Wireless LAN, Advanced Wireless LAN, Dr. Song Guo (Ed.), ISBN: 978-953-51-0645-6, InTech, Available from: http://www.intechopen.com/books/advanced-wireless-lan/qos-adaptation-for-realizing-interaction-between-virtual-and-real-worlds-through-wireless-lan

INTECH
open science | open minds