QoS Adaptation for Realizing Interaction between Virtual and Real Worlds in Pervasive Network Environment *

Shinya Yamamoto, Yoshihiro Murata, Naoki Shibata†, Keiichi Yasumoto and Minoru Ito
Graduate School of Information Science, Nara Institute of Science and Technology
Ikoma, Nara 630-0192, Japan
† Department of Information Processing and Management, Shiga University
(shiny-ya,yosihi-m,yasumoto,ito)@is.naist.jp, shibata@biwako.shiga-u.ac.jp

ABSTRACT

We propose a framework called FAIRVIEW which realizes cooperative work and interaction between mobile users in real world and remote network users. FAIRVIEW allows mobile users and network users to share the same view of a shared space including many moving objects in sufficient quality for interaction. For this purpose, we devised a mechanism which continuously measures information (called AR information) of the position and direction of each object and delivers AR information to user terminals, so that users can see moving objects in real-time. In order to realize real-time delivery of AR information with limited network resource in an ordinary wireless LAN and Internet environment, we propose a QoS adaptation mechanism which allows users to observe more important objects with a higher framerate. Through experiments supposing network environments with various available bandwidths, we confirmed that the proposed mechanism achieves a practical framerate of moving objects between mobile users and remote users in pervasive network environment.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications

General Terms

algorithms

Keywords

Networked Virtual Environment, Interaction, QoS Adaptation, Augmented Reality

1. INTRODUCTION

Recently, there are many studies regarding to MR (Mixed Reality) and AR (Augmented Reality)[6, 8, 18]. Many research efforts have also been made for NVE (Networked Virtual Environment) and CSCW (Computer Supported Cooperative Work) [7, 17]. These technologies allow remote users to participate in social activities such as shopping, exhibition, sports, and game which are held in real space. For realizing DVE (Distributed Virtual Environment), 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 [1, 4, 13] and NVEs for remote cooperation [11, 12]. 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 paper, we propose a framework named FAIRVIEW 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 PDAs, 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 the object is displayed as a 3D graphics on the display of virtual user terminal. To achieve the criterion (5), we propose a mechanism for delivering AR information as well as action/reaction to object in real time (which we call *AR event delivery mechanism*, hereafter) to realize smooth cooperation among real and virtual users. 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

*This work was partly supported by Hayao Nakayama Foundation.

of objects. 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

In [1, 4, 13], efficient AOI (Area of Interest) management methods are proposed. In [13], game space is dynamically divided based on Voronoi diagram for direct communication among players in a same fragment. In [4], 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 [1], the shared virtual space is divided into honeycomb regions, and a mechanism based on Pastry[2] 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 [11, 12], a load distribution and a QoS adaptation mechanisms for DVEs are proposed, respectively. In [12], 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 [11], 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[14] 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]. The main functions of FAIRVIEW are as follows: *Function (1)*

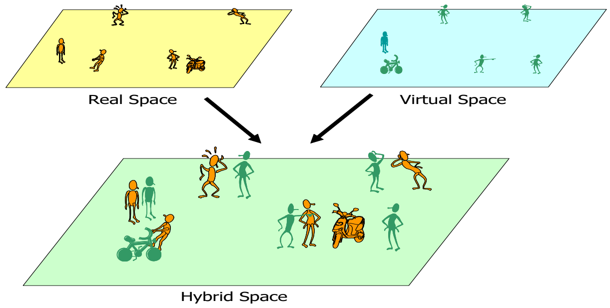

Figure 1: Hybrid Space Produced by FAIRVIEW

Providing the same view to real/virtual users: Virtual users can change their positions in virtual space using keyboards and mice 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 [16]. *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 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.

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. Players can be added in the case when there are too few real users. 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 PDA, and HMD

[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.

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

Table 1: User Equipment

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

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 softwares like ARToolkit [18] for positioning and detecting orientations of objects. Virtual users use ordinary PCs for FAIRVIEW.

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 [14], 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 outside of building. If a user is inside building, we use an indoor positioning method such as a method using wireless LAN access points[9], a method using speakers and microphones [5], Place Lab [19], and Weavy [10]. The orientation of an object can be measured using rotational and translational acceleration sensors, methods based on image processing such as ARToolkit [18] or the method in [6]. 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 paper. 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. As a solution, we present an AR event delivery mechanism in Sect. 4, and a QoS adaptation mechanism in Sect. 5.

4. AR EVENT DELIVERY MECHANISM

This section describes details of 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.

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.

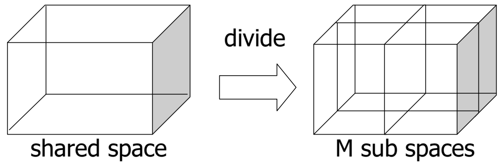

Figure 2: Division of Shared Space

For each o in $RO \cup VO$, let $AR(o) = (pos, angleH, angleV)$ be the AR information of o, where pos, $angleH$, and $angleV$ are the position on or beyond H, the horizontal angle to $hvec$, and the vertical angle to $vvec$, of object o, respectively. Each item of AR information is referred to by e.g., $o.pos$. We suppose that each real user terminal in RN can measure its user's AR information at 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 coverd 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 information exchange. The data including AR information exchanged among user nodes is called *AR event*, hereafter. In order to reduce processing and traffic load of each server node based on user's AOI, we 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 [15]. 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 two types of new server nodes called *object management node (om-node)* and *bandwidth controller node (bwc-node)*.

One om-node is prepared for each virtual object except for virtual users [3]. Let $on(o)$ be the om-node for virtual object o in $VO - VU$. Then $on(o)$ produces the reaction of o when a user or an object takes an action (e.g., pushing, holding) to o. When $on(o)$ receives an action event from a user node, $on(o)$ calculates o's reaction with physical simulation, and delivers a series of o's AR events to users who are observing o.

One bwc-node is prepared for each virtual user terminal vn in

[3]We do not prepare om-node for real object, since only real users can take actions to real objects and reaction can be observed physically or virtually through AR information measured for the object.

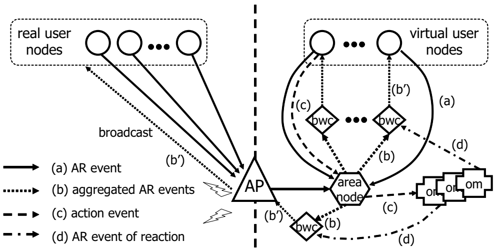

Figure 3: Overlay Network for AR Event Delivery

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[4]. 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 the latest appearance of objects on display of $node(u)$. To receive AR event, $node(u)$ uses publish/subscribe model [3]. 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)). Then an_A forwards the event to o's om-node $on(o)$, and $on(o)$ calculates the reaction and sends the AR events via the bwc-node to the users who are observing o (Fig. 3 (d)).

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 [5].

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.

Object Management Node The om-node $on(o)$ manages the reaction to o. One node is assigned to each object, but one node may manage multiple objects.

For the action taken to virtual object o, $on(o)$ calculates the reaction with physical simulation and sends AR events for the reaction to o's watcher 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]In FAIRVIEW, due to wireless bandwidth limitation, we restrict all real users to watching each virtual object at the same framerate.

[5]The node selection/replacement algorithm for area nodes, bwc-nodes, and om-nodes are omitted due to the space limitation. We use a similar technique to [15].

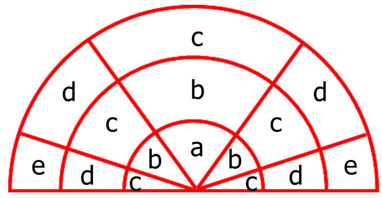

Figure 4: User's View and Zones with Relative Importance

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

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

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 ditributed 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.

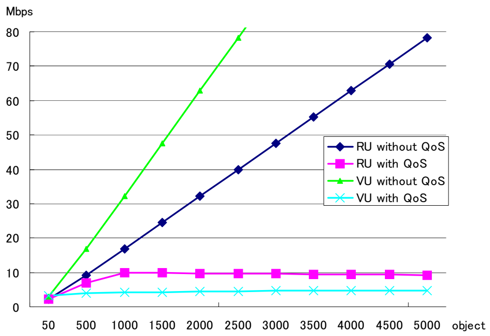

Figure 5: Required Bandwidth for User Terminals

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 $\frac{10\times 10Mbps}{10+25+15} = 2Mbps$, $\frac{25\times 10Mbps}{10+25+15} = 5Mbps$, and $\frac{15\times 10Mbps}{10+25+15} = 3Mbps$ 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 set the importance value $Imp_r(o)$ of virtual object o in VO observed by real users to the maximum value among the users observing this object. That is, we define that $Imp_r(o) = Max_{u\in Watcher(o)}(Imp(o, u))$.

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 measured the required bandwidth for application of users and objects. We also measured the framerates at which users can watch the objects for two cases with and without our QoS adaptation mechanism. We also evaluated user satisfaction by questionnaire.

6.1 Configurations

Experimental configurations are as follows. The sizes of virtual space V and real space R are both 50×50 m. There is one wireless AP whose radio range covers the whole real space R. The available wireless bandwidth BW_{AP} is 10Mbps, and all real users share this bandwidth. One area node and bwc-nodes are allocated on PCs on fixed wired network. The available bandwidth $bw_{AP}(vn)$ between each PC vn and the AP is 3Mbps. 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 om-nodes are not used in this experiment. 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. We also set the importance values for zones a, b, c, d, and e to 64, 16, 4, 2, and 1, respectively, since the difference between a and b should be larger than that among c, d, and e.

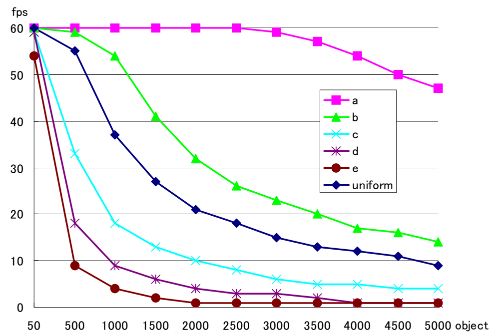

Figure 6: Framerates Obtained by Virtual Users

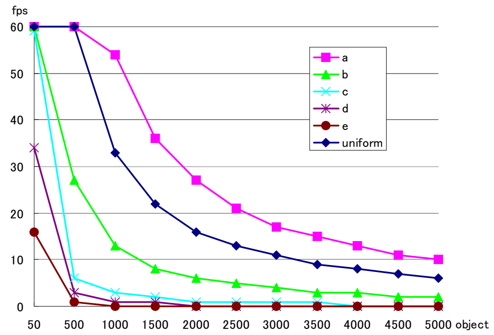

Figure 7: Framerates Obtained by Real Users

6.2 Required Bandwidth for User Terminals

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).

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.

Table 2: Subjective Evaluation by Questionnaire

Type		fps	ratio of right answer	comprehensibility	time to answer
With QoS	a	24	100%	4.5	5.2 sec
	b	7	100%	3.5	7.2 sec
	c	2	83%	2.0	10.5 sec
uniform	a	4	100%	3.3	10.7 sec
	b	4	83%	3.0	6.8 sec
	c	4	66%	2.3	11.1 sec

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. 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 Evaluation by Questionnaire

We evaluated the proposed method by questionnaire based on the results above. We prepared movies in which a person draws gesture of one of the following shapes: circle, spiral, triangle heading upward, triangle heading downward, square, diamond, heart, star and arrow. We produced composite movies by placing these movies in view zones a, b, and c, and setting their picture sizes and framerates to 368x488/24fps, 184x244/7fps, and 92x122/2fps. We also prepared a composite movie containing the three movies with the same framerate of 4fps. These framerates are decided based on the case when the number of objects is 5000, as shown in Fig. 6.

We asked five testees to see the movies and measured the time to recognize what the person in the movie is drawing, subjective comprehensibility in five levels (larger is better), and the ratio of correct answers. The results in Table 2 show that the testees recognized the important objects more accurately and quickly with our QoS adaptation method.

7. CONCLUSION

In this paper, we proposed a framework for interaction between real and virtual users in hybrid shared space, and a QoS adaptation mechanism for implementation in a network with bandwidth limitation. We confirmed that our method can handle hundreds of real and virtual users and thousands of objects with sufficient framerate in an ordinary wireless LAN and internet environment.

In this paper, we only coped with bandwidth limitation, but we are planning to extend our method to guarantee short latency.

We will also enhance our method assuming that bwc-nodes send control packets to upstream so that they give feedback of QoS adaptation.

8. REFERENCES

[1] A. P. Yu and S. T. Vuong: "MOPAR: A Mobile Peer-to-Peer Overlay Architecture for Interest Management of Massively Multiplayer Online Games," Proc. of ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV '05), (2005).

[2] A. Rowstron and P. Druschel: "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," Proc. of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp. 329-350, (2001).

[3] A. S. Tanenbaum and M. V. Steen: "Distributed Systems - Principles and Paradigms," Prentice Hall, (2002).

[4] B. D. Vleeschauwer, B. V. D. Bossche, T. Verdickt, F. D. Turck, B. Dhoedt, P. Demeester: "Dynamic Microcell Assignment for Massively Multiplayer Online Gaming," Proc. of 4th ACM Workshop on Network and System Support for Games (NETGAMES '05), (2005).

[5] J. Scott and B. Dragovic: "Audio Location: Accurate Low-Cost Location Sensing," Proc. of 3rd International Conference on Pervasive Computing(Pervasive 2005), LNCS 3468, (2005).

[6] K. Fudono, T Sato. and N. Yokoya: "Interactive 3-D modeling system using a hand-held video camera," Proc. of 14th Scandinavian Conference on Image Analysis (SCIA2005), pp.1248-1258, (2005).

[7] M. Fujimoto and Y. Ishibashi: "Packetization Interval of Haptic Media in Networked Virtual Environments," Proc. of 4th ACM Workshop on Network and System Support for Games (NETGAMES '05), (2005).

[8] R. Ichikari, K. Kawano, A. Kimura, F. Shibata and H. Tamura: "Mixed Reality Pre-visualization and Camera-Work Authoring in Filmmaking," Proc. of 5th International Symposium on Mixed and Augmented Reality, pp.239-240, (2006).

[9] T. Kitasuka, T. Nakanishi and A. Fukuda: "Wireless LAN based Indoor Positioning System WiPS and Its Simulation," Proc. of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03), pp.272-275, (2003).

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

[11] M. Eraslan, N. D. Georganas, J. R. Gallardo and D. Makrakis: "A Scalable Network Architecture for Distributed Virtual Environments with Dynamic QoS over IPv6," Proc. of IEEE Symposium on Computers and Communications (ISCC'2003), (2003).

[12] R. Chertov and S. Fahmy: "Optimistic Load Balancing in a Distributed Virtual Environment," Proc. of ACM Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '06), (2006).

[13] S. Y. Hu, G. M. Liao: "Scalable peer-to-peer networked virtual environment," Proc. of ACM 3rd Workshop on Network and System Support for Games (NETGAMES '04), pp. 129–133, (2004).

[14] Z. Yang, Y. Cui, B. Yu, J. Liang, K. Nshrstedt, S. Jung and R. Bajscy: "TEEVE: The Next Generation Architecture for Tele-Immersive Environments," Proc. of 7th IEEE International Symposium on Multimedia (ISM'05), (2005).

[15] S. Yamamoto, Y. Murata, K. Yasumoto and M. Ito: "A Distributed Event Delivery Method with Load Balancing for MMORPG," Proc. of 4th ACM Workshop on Network and System Support for Games (NETGAMES '05), (2005).

[16] K. Yasumoto and K. Nahrstedt: "Realistic Voice Chat Framework for Cooperative Virtual Spaces," Proc. of IEEE 2005 International Conference on Multimedia and Expo (ICME 2005), CD-ROM, (2005).

[17] Z. Yang, B. Yu, K. Nahrstedt and R. Bajscy: "A Multi-stream Adaptation Framework for Bandwidth Management in 3D Tele-immersion," Proc. of ACM Network and Operating System Support for Digital Audio and Video (NOSSDAV' 06), (2006).

[18] ARtoolkit, http://www.hitl.washington.edu/artoolkit/.

[19] Intel Research: Place Lab, http://www.placelab.org/.