Immersive Voice Communication for Massively Multiplayer Online Games

Jehn-Ruey Jiang, Jih-Wei Wu, Chi-Wen Fan, and Jie-Yi Wu

Abstract In this paper, we propose a voice communication scheme, called *immersive voice communication (IVC)*, to provide *massively multiplayer online game (MMOG)* players with the immersive experience to hear the voice of neighbor players within the *area of interest (AOI)*. IVC is a peer-to-peer based scheme, so it does not impose too many extra loads on the original MMOG system. It further uses a relation model to classify neighbors of a player into *listeners* and the *overhearers*, and allocates less bandwidth to the latter than to the former for reducing the voice data traffic without sacrificing the user experience. IVC is also a latency- and bandwidth-aware scheme. It adopts the *network coordinate system (NCS)* to help construct the *adaptive k-ary tree (Ak-tree)* to reduce the voice data transmission latency and efficiently utilize the bandwidth. As shown by the simulation results, the proposed scheme outperforms other related schemes. We also implement IVC and integrate it with a *spatialized voice rendering* mechanism to realize an MMOG-like gallery guiding system for evaluating the user experience of IVC under the *technology acceptance model (TAM)*. The TAM analysis results show that users consider IVC helpful and easy to use, and thus have high intention to use IVC.

Keyword: peer-to-peer; immersive voice communication; massively multiplayer online games; spatialized voice rendering; technology acceptance model

J. R. Jiang, J. W. Wu, C. W. Fan, and J. Y. Wu
Department of Computer Science and Information Engineering, National Central University, Taiwan, Republic of China

J. R. Jiang (✉)
e-mail: jrjiang@csie.ncu.edu.tw

J. W. Wu
e-mail: jihwei.wu@gmail.com

C. W. Fan
e-mail: allanfann@gmail.com

J. Y. Wu
e-mail: pokkbaby@msn.com

1 Introduction

Massively multiplayer online games (MMOGs), a special genre of *networked virtual environments (NVEs)*, have been one of the most popular applications on Internet nowadays. More and more people register themselves as MMOG users or *players*, and go online to play games concurrently. For example, the World of Warcarft [33], one of the most popular and successful MMOGs, has had more than 12 million registered players since it was released in 2004. The players, assuming the representation of virtual *avatars*, interact or socialize with one another to have fun in a virtual world or virtual environment which is synthesized by computer graphics technologies.

Text chat is the most common way for MMOG players, during playing games, to exchange messages by keyboard typing. However, it will distract players from playing the game. With richer and richer game content media and gameplay styles, the voice chat may be a better way of communication in MMOGs. This motivates us to deceive an immersive voice communication scheme for MMOG users to conversate or to just overhear other users conversate.

Most MMOGs are equipped with brilliant pre-recorded, locally-stored sound effects and sound tracks to increase the gameplay experience. However, not all MMOGs support voice communication mechanisms for players to talk to each other. Even though several games have provided in-game voice chatting utilities, the scale of voice chatting groups is usually limited. For example, the World of Warcarft has a built-in voice communication system, in which a player can join only one preselected chatting channel that can accommodate at most 40 users. The reason for the small scale of chatting groups is that the voice transmission consumes much more bandwidth than general operations for playing a game. Currently, most MMOGs are based on the client/server architecture, which has the inherent limitation on scalability due to bounded available network bandwidth and computing resources. In such an architecture, additional voice transmission will seriously degrade the scalability of the MMOGs. Third-party voice communication software, such as TeamSpeak [28], Skype [27] and Ventrilo [32], may be a

transitional solution to meet the needs of voice communication in MMOGs. However, this kind of software is based on the server/client architecture and thus suffers from the scalability problem. Furthermore, when using such software, users have to manually solicit members of a collaborative team or a special community to form a conversation group before the game starts. These third-party voice communication programs also lack the mechanism to synchronize with the states of games. When the game state changes, the users thus have to manually adapt their conversation group to fit into the dynamic game conditions.

The immersive voice or the spatialized voice (sound) [2] could raise the quality of user experience by integrating voices into visual display and user interactive operations in virtual-environment applications. As mentioned above, current MMOG systems do not provide regular voice communication, not to mention the immersive voice communication. Few immersive audio communication solutions [26] [34] are designed for MMOGs or NVEs; most of them employed complicated centralized audio rendering and/or heavy-duty audio mixing schemes, reducing the feasibility to integrate the immersive voice communication service into current MMOGs. Therefore, we need to conceive a brand new immersive voice communication mechanism for players in an MMOG to talk to one another easily without bringing too many extra loads to the MMOG system.

This paper proposes an immersive voice communication scheme, called *immersive voice communication (IVC)*, to provide MMOG players with the immersive experience to hear the voice of neighbor players within the *area of interest (AOI)*. For a player, its AOI is the area of a fixed-radius circle centered at it, and it usually interacts with and visualizes only players within the area. To prevent the voice messages from consuming exceeded access bandwidth of peers, IVC exploits a voice delivery architecture derived from human auditory sensation. By observing human behavior, we have found that a person usually focuses on a specific voice and ignore other voices. Therefore, the AOI neighbors of a player are classified into two categories: the *listener*, who focuses on listening to the player's voice, and the *overhearer*, who just overhears the player's voice. A player can allocate less bandwidth to the overhearer than the listener, since a listener requires better quality of voice but an overhearer does not. This can be utilized to reduce the voice data traffic dramatically. We further develop a *relation model* for a player to determine to listen to which neighbor based on the criteria such as the distance, relative orientation, and social relationship between any pair of players within the AOI.

IVC is a *peer-to-peer (P2P)* based scheme. It can be integrated into traditional client/server-based MMOGs [33] to render them with the voice communication ability without causing too many additional loads to them. Certainly, IVC can also be integrated into P2P-based MMOGs [12-13]. In a P2P scheme, a participating *peer* (or *node* or *player*) not only consumes resources, but also contributes resources, such as the computing capacity, network bandwidth and storage space. The MMOG servers therefore do not bear too many extra loads while the number of participating peers increases.

IVC is both bandwidth- and latency-aware, as explained below. It integrates the ***Ak-tree*** (adaptive k-ary tree) algorithm, which evolves from the LGK (location guided k-ary tree) algorithm [5], to construct a multicast tree with the smallest number of levels, in which the speaking peer is the root node of the multicast tree, and its voice data are transmitted to all nodes on the tree in a recursive parent-to-children manner. In the Ak-tree algorithm, each peer first evaluates its available outgoing bandwidth and then takes as many as possible child nodes to form the multicast tree to completely utilize the outgoing bandwidth and reduce the queuing delay caused by insufficient outgoing bandwidth for outward data, thus reducing the voice data dropping rate. IVC is thus bandwidth-aware.

IVC also uses the Vivaldi *network coordinate system (NCS)* [9] to reduce the voice data transmission latency. Every peer has an associated NCS coordinate and the *NCS distance* of two peers (i.e., the Euclidean distance between the NCS coordinates of the two peers) implies the transmission latency between them. The closer two nodes in the NCS are, the smaller the latency between the two nodes is. The Ak-tree algorithm first considers the speaking peer's listeners and attaches the listeners of shorter transmission latency to the Ak-tree preferentially. The algorithm also allocates more bandwidth to listeners than to overhearers. After allocating bandwidth to all listeners of a speaking peer, the Ak-tree algorithm then considers the speaking peer's overhearers and attaches the overhearers of shorter transmission latency to the Ak-tree preferentially. IVC is thus latency-aware.

We evaluate the performance of IVC by simulation experiments and compare the results with those of related ones. We also implement IVC and integrate it with a simple *spatialized voice rendering* mechanism to render player voices with general binaural audio devices. This binaural spatialized voice rendering mechanism is easy to implement via common audio SDKs, like Microsoft Direct Sound and OpenAL, designed for the off-the-shelf audio hardware of PCs or ordinary mobile devices. We further build an

MMOG-like virtual gallery guiding system using IVC and the spatialized voice rendering mechanism to evaluate if IVC can be accepted by MMOG users with the *technology acceptance model (TAM)*. The TAM analysis results show that MMOG users have high intention to use IVC due to its helpfulness and ease to use.

The rest of this paper is organized as follows. Section 2 describes some related work. Section 3 proposes the IVC scheme, and Section 4 proposes the spatialized voice rendering mechanism. In Section 5, we show the simulation results of IVC and compare them with those of related schemes. We demonstrate user acceptance analysis of IVC in Section 6, and finally conclude the paper with Section 7.

2 Related Work

In this section, we review some research results related to the IVC scheme, including spatialized sound rendering for virtual environments, and distributed multicast algorithms.

2.1 Spatialized sound rendering model for virtual environments

With the availability of powerful hardware, virtual environments have steadily gained popularity over the past decade. Not only exquisite visual display is regarded as essential for virtual environments, but immersive sound effects are also considered to be crucial enhancement for the experience of use. One way to realize immersive sound effects is through spatialized sound rendering. Some spatialized sound rendering techniques were introduced to accurately reproduce immersive sound in the applications of virtual environments, theaters, and multi-media entertainments. These spatialized sound rendering schemes employ sophisticated auditory equipments and technologies, such as the Hi-Fi two-channel stereophony, multichannel speaker array, and 3D surrounding audio system, to reproduce spatial attributes of sound (e.g., the direction, distance, width of sound sources, and room envelopment) to the sound listener.

Begault [2] proposed a systematic source-medium-receiver model to study the spatialized sound rendering in immersive virtual environments. He pointed out that the distance and angle between a sound source and a sound listener are the most important attributes of the sound when the sound is to be reproduced in virtual environments. Pulkki [25] proposed Vector-Based Amplitude Panning (VBAP), a method for generating amplitude-panning virtual sounds pointing to arbitrary directions in a 2D or 3D sound field with any number of loudspeakers placed arbitrarily. The virtual source of the sound can be regarded as on the sector defined by the locations of the loudspeakers and the sound listener by controlling the loudspeakers' sound amplitudes. When the number of loudspeakers increases, the virtual sound source can be more precisely localized by the sound listener. However, the limitation of VBAP is that the position of sound listeners is assumed to be known apriori, fixed and restricted to a small area [19]. Distance-Based Amplitude Panning (DBAP) [23] makes no assumptions about where the sound listeners are situated, and does not need to know the loudspeaker arrangement in advance. DBAP uses the distance between the virtual sound source and the sound listener to calculate the gains, and the gain for each loudspeaker is independent of the listener's position. Comparisons in [19] show that DBAP surpasses VBAP in terms of computational load, positioning accuracy, flexibility and simplicity.

The aforementioned spatialized sound rendering systems rely on multichannel speaker auditory devices or a 3D speaker array to improve the positioning accuracy for immersive virtual environments [10][24]. However, we cannot assume such specifically designed auditory devices in the virtual environments of MMOGs, since most MMOG players use general-purpose stereo loudspeakers or headphones as their auditory devices. Head Related Transfer Function (HRTF) [25] is a binaural sound generating technique that can use only 2 loudspeakers or a pair of headphones to synthesize a 3D sound field. It characterizes how each ear of a listener receives sound information from a sound source in space. The sound strength which each ear hears is measured and stored as the Head Related Impulse Response (HRIR) for the sound source at every possible position. The corresponding HRTF values retrieved from the HRIR database for each ear can then be used to synthesize a binaural sound source before being played back on loudspeakers or headphones [25]. However, the overheads and the computational load of the HRTF rendering model are much higher than those of VBAP and DBAP.

2.2 The immersive voice communication in MMOGs

By using immersive voice communication, users can communicate, interact and collaborate with each other over a distance with immersive and lifelike experiences [1]. However, to implement immersive voice communication in MMOGs or NVEs will face more challenges than in the prearranged spatial audio applications, due to the dynamic nature of networked environments and the limited network bandwidth. To overcome the challenges, many P2P schemes

are proposed to disseminate voice data by constructing multicast tree among peers. Since the core of those schemes is the construction of the multicast tree rooted at the peer (or node) which is the source of a voice, we introduce only the multicast tree construction concept (or algorithm) of each scheme, and the readers are referred to the original paper for the scheme's implementation details.

To the best of our knowledge, all P2P schemes, except IDM (*indirect dissemination mechanism*) [21], construct intra-AOI multicast trees which contain only AOI neighbors of the root node. Liang et al. proposed IDM [21], which is extended form their own previous studies [22][35], to construct the multicast tree to include the root node's AOI neighbors and non-AOI neighbors (i.e., the nodes outside the root node's AOI) for spatialized voice communication. Since multicast schemes using non-AOI neighbors require the non-AOI neighbors to receive and forward voice data not destined to them, it is likely that non-AOI neighbors will not forward others' data if no incentive is offered. For such cases, the multicast schemes may not work well. Therefore, we introduce only intra-AOI multicast tree construction schemes below.

The QuadCast and SectorCast schemes proposed by Jiang et al. [18] are effective AOI voice data multicasting mechanisms based on P2P networking. Both schemes construct a multicast tree to enable an MMOG player to send voice data to all its neighbor players within AOI. The MMOG world is modeled as a 2D plane, and each player, a point on the plane whose coordinate is known to all its AOI neighbors. In QuadCast, the player sets its coordinate as the origin and partitions its communication zone into four quadrants. The nearest peer in each quadrant is selected as the child peer of the player. Each child peer gets voice data and a recipient list of peers from its parent peer to forward the data. Each peer repeats such quadrant partition and the child-peer selection procedure until all peers are on the multicast tree. Fig. 1(a) shows an example of QuadCast multicast tree construction. In Fig. 1(a), the player A chooses the nearest peers B, C, G, and I as its child peers in each quadrant; B, C, G, and I then select their respective child peers until all peers are on the multicast tree.

QuadCast partitions the communication zone into four quadrants. For some cases, the number of peers in quadrants might vary dramatically. The subtree in a quadrant with more peers usually has more levels, which leads to larger transmission latency. To overcome the problem, SectorCast partitions the communication zone into sectors (or subzones) according to the number of peers within every subzone. By sorting peers according to their polar angles and scanning them clockwise, the subzones in SectorCast have about the same number of peers and the latency is further reduced.

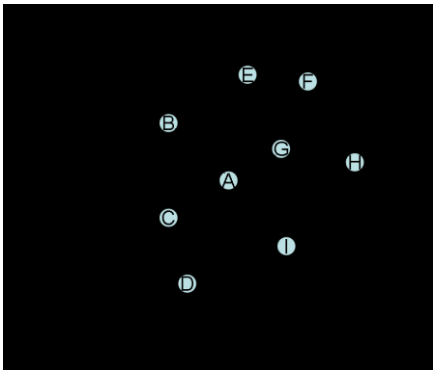

Fig. 1(a). An example of the QuadCast scheme

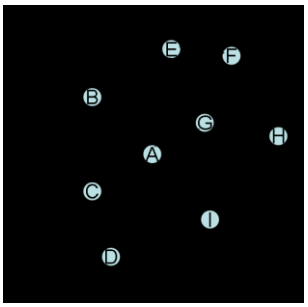

Fig. 1(b). An example of the SectorCast scheme

It is worth mentioning that both QuadCast and SectorCast employ the voice data packet aggregation technique to reduce the network traffic. The peers repack the voice data packets with the same destinations into an aggregated packet, which not only eliminates duplicated packet headers but also lightens the routing workload. QuadCast and SectorCast also employ the voice-mixing technique to further reduce the network traffic. However, the schemes have the problem that users' immersive experience may be lessened since the mixed voice prevents a user from differentiating the orientation and position of each original voice.

In [29], Triebel et al. also proposed a P2P voice communication scheme for MMOGs. The scheme transmits voice data via the following two types of communication. (1) Location-based communication: The scheme sets the speaking player's location in the virtual world as the center of a circle, and partitions the player's surrounding area into 4 levels of concentric circles, namely the private zone, social zone, public zone and world zone, in the order from the center to the outward. The more inner circle the players are on, the better quality of voice they should receive from the speaking player. (2) Group-based communication: The speaking player

puts other players with special relationship, such as partnership or teammateship, into a special recipient list. The speaking player then sends voice data with stable quality to players in the list to maintain the special relationship.

Below, we introduce a related multicast algorithm, called the LGK (Location-Guided k-ary tree) algorithm, which is proposed in [5] to provide group communication in mobile ad hoc networks (MANETs). Although the LGK algorithm is intended to be run in MANETs, its concept can be borrowed to construct multicast trees for AOI voice communication in MMOGs. The LGK algorithm assumes the location of each participating node is known and constructs a k-ary multicast tree with the following two steps for a node to perform group communication to send data to all participating nodes (i.e., all group members). The LGK algorithm regards the node initiating the group communication as the root node and executes the following two steps. (Step 1) The root node selects the closest k nodes as its child nodes. (Step 2) The nodes not selected are grouped into k clusters according to the geometric proximity to the k child nodes. Each child node then recursively takes itself as the root node and executes the above two steps in the corresponding cluster to form a subtree. The recursion stops when the root node has no node to select. In this recursive manner, the k-ary tree is formed.

Fig. 2 illustrates the LGK algorithm execution for the case of k=2. In Fig. 2, peer A selects the 2 nearest neighbors B and F as its child nodes; B and F then select their respective child nodes to form the final 2-ary (or binary) multicast tree.

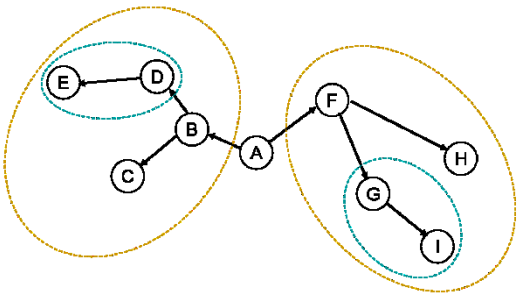

Fig. 2. An example of the multicast tree construction of the LGK algorithm for k=2

3 The IVC Scheme

In this section, we describe the listening relation model and the adaptive k-ary multicast tree construction algorithm of the proposed *immersive voice communication (IVC)* scheme.

3.1 Listening relation model

In real life, people usually pay their attention only to some specific things that they care for [8]; the other things are relatively unimportant or even trifling to them. According to the degree of attention that a person pays to voices, we classify voices into two categories: *listening voice* and *overhearing voice*. If a person *A* pays attention to a speaker *B*, then *B*'s voice is a listening voice to *A* and *A* is a *listener* of *B*. We assume a person can focus on a sole voice at one time. Therefore, *B*'s voice is the sole listening voice to *A*, other voices are overhearing voices to *A*, and *A* is an *overhearer* of speakers that are not *B*.

In IVC, the listening voice is transmitted with much higher bandwidth than the overhearing voice so that the network traffic can be reduced without compromising users' listening experience. IVC uses a *listening relation model* to periodically score a player i's all *AOI neighbors* (i.e., the players within i's AOI). If a neighbor j has the highest score, then player i sends j a "Subscribe" message to notify j that i is j's listener (i.e., i focuses on only j's voice). Implicitly, player i is an overhearer of those neighbors not receiving the "Subscribe" message from i. Later on, if another neighbor h becomes the one with the highest score, then i sends j an "Unsubscribe" message and sends h a "Subscribe" message. Henceforth, i becomes h's listener and j's overhearer.

The listening relation model scores every AOI neighbor j of player i with three subscores $R_{i,j}^{dist}$, $R_{i,j}^{ori}$, and $R_{i,j}^{social}$ in the aspects of distance, orientation and social relationship, as described below.

a) Distance:

$$R_{i,j}^{dist} = 1 - \frac{dist_{i,j}}{dist_{max}}, \quad (1)$$

where $dist_{max}$ is the maximal distance (i.e., the radius of AOI) at which i can see players and $dist_{i,j}$ is the distance between player i and player j. It is noted that the subscore $R_{i,j}^{dist}$ ranges between 0 and 1. In general, the closer j is to i, the larger the subscore is.

b) Orientation:

$$R_{i,j}^{ori} = \frac{\cos^{-1}(Ori_i \cdot Ori_j)}{\pi}, \quad (2)$$

where Ori_i and Ori_j are the unit-length orientation vectors of player i and player j, respectively, and $Ori_i \cdot Ori_j$ stands for the inner product of Ori_i and Ori_j. Note that the inner product of two vectors equals to the multiplication of the first vector length, the second vector length, and *cos* θ,

where θ ($0 \le \theta \le \pi$) is the angle between the two vectors. Since Ori_i and Ori_j are unit-length vectors, the subscore $R_{i,j}^{ori}$ is actually the ratio of the angle between Ori_i and Ori_j over π; $R_{i,j}^{ori}$ ranges between 0 and 1. The angle between Ori_i and Ori_j is between 0 and π. When players i and j are face to face, the angle is π and the subscore is 1; when players i and j face to the same direction, the angle is 0 and the subscore is 0. People usually talk to each other face to face, so the larger the subscore is, the more likely they are talking to and listening to each other.

It is noted that the subscore $R_{i,j}^{ori}$ will be 1 when players i and j are face to face or when they are back to back. The subscore therefore should be adjusted according to the positions of players i and j. For example, if player i's orientation is towards the quadrant 1 and player j is also in the quadrant 1, then players i and j are indeed face to face and the subscore should be 1. However, if player i's orientation is towards the quadrant 1 and player j is in the quadrant 3, then players i and j are indeed back to back and the subscore should be 0.

To adjust the subscore $R_{i,j}^{ori}$, we assume only the AOI neighbors of player i within the *covering range* of $Ori_i+\pi/2$ and $Ori_i-\pi/2$ can hear player i's voice. Similarly, we assume player j can only hear the voice from its AOI neighbors within the *hearing range* of $Ori_j+\pi/2$ and $Ori_j-\pi/2$. The subscore $R_{i,j}^{ori}$ will be adjusted to be 0 if j is not within i's covering range or if i is not within j's hearing range.

c) Social relationship:

$$R_{i,j}^{social} = \begin{cases} 1, \text{if player i has social relations with player j} \\ 0, \text{otherwise} \end{cases} \quad (3)$$

People usually prefer listening to the voices from their friends, partners, or teammates in a game, that is, those who have social relations with them. Therefore, if player j has some specific relations with player i (e.g., j is i's friend in a social network, or j is i's partners or teammates in games), then j will get score 1; otherwise score 0, as defined in Eq. (3).

After deriving the subscores $R_{i,j}^{dist}$, $R_{i,j}^{ori}$, and $R_{i,j}^{social}$, we can get the final listening relation score $R_{i,j}$ for every AOI neighbor j of player i by combining $R_{i,j}^{dist}$, $R_{i,j}^{ori}$, and $R_{i,j}^{social}$ with respective weights, w^{dist}, w^{ori} and w^{social}, as shown in Eq. (4), where $0 \le w^{dist}, w^{ori}, w^{social} \le 1$, and $w^{dist}+w^{ori}+w^{social}=1$.

$$R_{i,j} = w^{dist} R_{i,j}^{dist} + w^{ori} R_{i,j}^{ori} + w^{social} R_{i,j}^{social} \quad (4)$$

3.2 Adaptive k-ary tree (Ak-tree) construction

The LGK tree algorithm [5] constructs a k-ary tree with a fixed k value for a node to perform multicast in a MANET. In light of the LGK algorithm, we propose an adaptive k-ary tree construction mechanism for a player or peer in IVC to directly deliver voice data to k of its AOI neighbors and then to indirectly and recursively deliver voice data to all of its AOI neighbors. As peers usually have varying uploading (or outgoing) bandwidth of delivering data, the k-ary tree with a fixed k value might not properly utilize the outgoing bandwidth of peers. In IVC, the parameter k is set adaptively according to a player's available bandwidth to efficiently utilize the bandwidth. IVC assumes that a peer can estimate its available outgoing bandwidth and reserve a reasonable part of it for IVC. Hence, the peer's *capacity* can then be determined properly, where the capacity is defined as the maximum number of basic voice data packets that the peer can deliver per unit-time.

In IVC, listeners receive voice of better quality and less latency than overhearers do. The overhearing voice is relatively unimportant and can even be discarded due to insufficient capacity of relay peers. Consequently, in the process of constructing the multicast tree for delivering voice to AOI neighbors, a peer always first considers its listeners. The outgoing bandwidth of the peer is first allocated to listeners and then to the overhearers. Observing the practical network environment, IVC assumes that a peer's incoming (downloading) bandwidth is much larger than its outgoing (uploading) bandwidth and that the packet discard is due to outgoing bandwidth insufficiency rather than incoming bandwidth insufficiency.

IVC also assumes every peer knows the NCS coordinates of itself and all its AOI neighbors. Note that the Euclidean distance between the NCS coordinates of two peers implies the transmission latency between them. IVC follows the procedures proposed in [9] to calculate the NCS coordinates, of which details are not shown here due to space limitation. Also note that the NSC coordinates are not related to the virtual environment coordinates. In short, NCS coordinates are used to decide the latency between two peers, while the virtual environment coordinates are used to decide the AOI neighbors of a peer.

In IVC, every player first constructs its adaptive k-ary tree (Ak-tree) for delivering its voice. Below we show the adaptive k-ary tree construction algorithm (called Ak-tree

algorithm for short) in IVC. The notations used in the algorithm are defined in Table 1.

Algorithm Ak-Tree

Input: p_r, a_r, S_L, S_O, where p_r is the voice source peer, a_r ($a_r \geq b_L$) is the available capacity of p_r, S_L is the set of listeners of p_r, and S_O is the set of overhearers of p_r

Output: Ak-tree T rooted at p_r

1: p_r inquires of its AOI neighbors $p_1,\ldots,p_n$ in S_L and S_O about their available capacities $a_1,\ldots,a_n$, and their NCS coordinates.
2: $T= \{p_r\}; T_L = \{p_r\}; T_O = \emptyset$; $d(p_r) = 0$;
3: While $((T_L \cup T_O) \neq \emptyset)$ and $(S_L \cup S_O) \neq \emptyset)$ do {
4: $<p_i,p_j> =$ Coalesce($\text{Argmin}_{<p_i,p_j>, p_i \in T_L, p_j \in S_L}$ $(d(p_i)+d(p_i,p_j))$, $\text{Argmin}_{<p_i,p_j>, p_i \in T_O, p_j \in S_L}$ $(d(p_i)+d(p_i,p_j))$, $\text{Argmin}_{<p_i,p_j>, p_i \in (T_L \cup T_O), p_j \in S_O}$ $(d(p_i)+d(p_i,p_j))$);
5: If ($p_i \in T_L$ and $p_j \in S_L$)
6: {$w_i = w_i - b_L$; Move p_j from S_L to T_L; }
7: Else {$w_i = w_i - b_O$; Move p_j from S_L or S_O to T_O;}
8: Add p_j into T as a child node of p_i;
9: $d(p_j) = d(p_i) + d(p_i,p_j)$
10: AdjustMember(T_L, T_O);
11: } //End of While Loop
12: Return Ak-tree T;

Table 1. Notations used in the Ak-tree algorithm to construct the Ak-tree rooted at p_r

p_r	The root of the Ak-tree
a_r	The available network capacity of p_r
$p_1, p_2,\ldots p_n$,	The AOI neighbors of p_r
$a_1, a_2,\ldots,a_n$	The available bandwidth capacity of $p_1,p_2,\ldots,p_n$
b_L	The bandwidth capacity needed by listeners
b_O	The bandwidth capacity needed by overhearers
T	The Ak-tree rooted at p_r
T_L	The set of peers in T offering listener connection
T_O	The set of peers in T offering overhearer connection
S_L	The set of listeners of p_r
S_O	The set of overhearers of p_r
$d(p_i)$	The accumulated NCS distance from p_r to p_i
$d(p_i, p_j)$	The NCS distance between p_i and p_j.

Note that in the Ak-tree algorithm, $\text{Argmin}_{<p_i,p_j>,\ldots}$ $(d(p_i)+d(p_i,p_j))$, will return the parameter (i.e., a 2-tuple $<p_i, p_j>$) that generates the minimum value of $d(p_i)+d(p_i,p_j)$, which stands for the minimum accumulated latency from the root p_r to the peer p_i; it will generate a null value if no such 2-tuple exists. Also note that Coalesce(arguments…) will return the first non-null expression. Still note that function AdjustMember(T_L, T_O) will remove from T_L and T_O all members that has the capacity less than b_O; the function will also move a member from T_L to T_O if its capacity is less than b_L but larger than b_O.

To deliver a player p_r's voice, an Ak-tree rooted at p_r is constructed dynamically for every voice streaming session of the player. The voice source p_r is assumed to have bandwidth capacity that can support at least one listener (by $a_r \geq b_L$). A voice streaming session can span a continuous voice deliver period or can span a specific time interval. It all depends on the configurations of the applications.

The Ak-tree algorithm considers listeners preferentially; it allocates more bandwidth to listeners and attaches listeners of shorter transmission latency to the Ak-tree earlier (by $\text{Argmin}_{<p_i,p_j>, p_i \in T_L, p_j \in S_L}$ $(d(p_i)+d(p_i,p_j))$). When no peers in the current Ak-tree can support listener connection, a listener of shorter latency can still be allocated overhearer bandwidth preferentially (by $\text{Argmin}_{<p_i,p_j>, p_i \in T_O, p_j \in S_L}$ $(d(p_i)+d(p_i,p_j))$). Finally, remaining bandwidth capacity is allocated to overhearers with shorter latency (by $\text{Argmin}_{<p_i,p_j>, p_i \in (T_L \cup T_O), p_j \in S_O}$ $(d(p_i)+d(p_i,p_j))$). When a peer is selected to be added into the Ak-tree and is allocated with the listener (resp., overhearer) bandwidth, it is moved from S_L (resp., S_L or S_O) to T_L (resp., T_O). However, the members of T_L and T_O will be adjusted by function AdjustMember(T_L, T_O) according to the relation of b_L and b_O and the available bandwidth capacities of members.

Note that the returned Ak-tree may not contain all AOI neighbors of p_r, especially when peers have small capacity. Fortunately, the number of listeners is in general smaller than that of overhearers and listeners receive voice of better quality preferentially. Hence, even if some AOI neighbors are excluded from the Ak-tree due to insufficient peer capacity, they are usually relatively unimportant overhearers and the exclusion just causes minor overall adverse effects.

Figure 3 illustrates how the Ak-tree algorithm constructs peer A's Ak-tree to span peers B,…,L, where peers are shown according to their NSC coordinates, F and I are listeners and others are overhearers. It is supposed that $d(A,B)<d(A,C)<\ldots<d(A,L)$ and the bandwidth capacity requirement is 2 units for listeners and 1 unit for overhearers. Initially, the Ak-tree contains only the root none, peer A. The algorithm first attaches F, a listener of A, to the Ak-tree by setting F as a child node of A. The available capacity a_A of peer A is adjusted from 7 to 5. Second, the algorithm sets I as a child node of A to attach I to the tree and adjusts the available capacity a_A of peer A from 5 to be 3. Note that peer I is not set as a child node of F since the accumulated NCS coordinate distance from I to A is shorter than that from I to F, and then to A. Afterwards, based on the order of the

accumulated NCS coordinate distance to A and residual capacity, the algorithm attaches overhearers B, C, D, E, H, K and L to the Ak-tree one by one. In this example, the constructed Ak-tree has the depth of 3.

Node	Capacity	Node	Capacity	Node	Capacity
A	7	E	3	I*	3
B	3	F*	4	J	3
C	2	G	3	K	5
D	4	H	6	L	7

*F and *I are listeners of A; others are overhearers

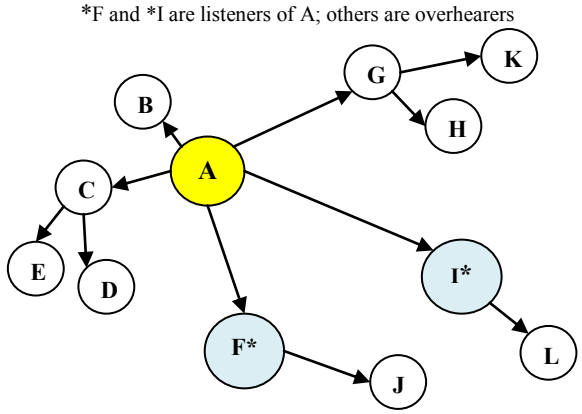

Fig. 3. An example of the Ak-tree construction for peers A,…, L, where peers are shown according to their NSC coordinates, A is the voice source (i.e., the root), and I and F are A's listeners.

4 Spatialized Voice Rendering

Rendering of spatialized sound is the reproduction of directional sound fields on audio devices. Some advanced rendering techniques use speaker arrays and ultrasonic speakers to render spatialized sound for high-quality video representation or free-view-point TV [1]. We suppose that most of MMOG users' computers are equipped with ordinary off-the-shelf audio devices, which generally are loudspeaker-pairs or headsets. Although VBAP, DBAP and HRTF techniques can be applied for spatial sound reproduction on general-purpose networked virtual environments, the heavy computing loads for reconstructing the sound fields in continuously changing scenarios make the techniques not suitable for some applications (e.g., MMOGs). We thus design a simplified spatialized voice rendering mechanism based on DBAP and integrate it into IVC. The mechanism consumes less computation power than its counterparts but performs a pretty good positioning effect of voice sources.

To render spatialized sound, DBAP considers the relative position of a sound source and a loudspeaker to calculate the loudspeaker gain to reconstruct the sound field [23]. We follow the concept of DBAP to design a simplified mechanism to render spatialized voices for a listening player in IVC. An MMOG assumes that a player is located at the center of AOI and can see and interact with all the avatars in the AOI. Similarly, IVC assumes the sound field is the AOI and the listening player is located at the center of the sound field (i.e., AOI) to calculate the gain according to Begault's research [2], which stated that the relative distance perception and the angular perception, namely the **azimuth angle** and the **elevation angle,** of a virtual sound (or voice) source can be used to locate the source in the virtual environment.

IVC assumes the sound field is the AOI and the listening player is located at the AOI center to calculate the distance d and the azimuth θ between a listening player and a voice source to get the gain (i.e., volume) of the voice, where the azimuth is the angle between the orientation of the player and the connection line of the source and the player (see Fig. 4). IVC does not consider the elevation angle of voice sources for the sake of simplicity and the limited computation power of ordinary user computers. The distance and angle calculation can be done by the listening player with the voice source's coordinate, which is carried in voice data packets, and the listening player's own coordinate and orientation. Let $d_{i,j}$ denote the distance between player i (i.e., the listening player) and player j (i.e., the voice source), and $\theta_{i,j}$ be the angle between player i and player j. With the assumption that the intensity of the voice is inversely proportional to the square of the propagation distance, we obtain the base voice volume $Volume_{base}$ according to Eq. (5).

$$Volume_{base} = Volume_{min} + (1 - (\frac{d_{i,j}}{r_{AOI}})^2) \times (Volume_{max} - Volume_{min}) \quad (5)$$

where r_{AOI} is the radius of AOI, $Volume_{max}$ is the maximum voice volume of the system, and $Volume_{min}$ is the least voice volume the player can hear when the voice source just locates on the edge of AOI.

IVC assumes the listening player is equipped with a left loudspeaker and a right loudspeaker. It defines ***VB*** (Voice Balance) in Eq. (6) as the factor of gains proportion for the two loudspeakers. The value of VB ranges from -1.0 to 1.0. When VB is positive (resp., negative), the right (resp., left) loudspeaker has the higher volume of $Volume_{right}$ (resp., $Volume_{left}$).

$$VB = cos\ \theta_{i,j} \quad (6)$$

Spatialized voice can be rendered by combining Eq. (5) and Eq. (6), as shown in Eq. (7).

$$\begin{cases} Volume_{left} = Volume_{base} \times \dfrac{1 - VB}{2} \\ Volume_{right} = Volume_{base} \times \dfrac{1 + VB}{2} \end{cases} \quad (7)$$

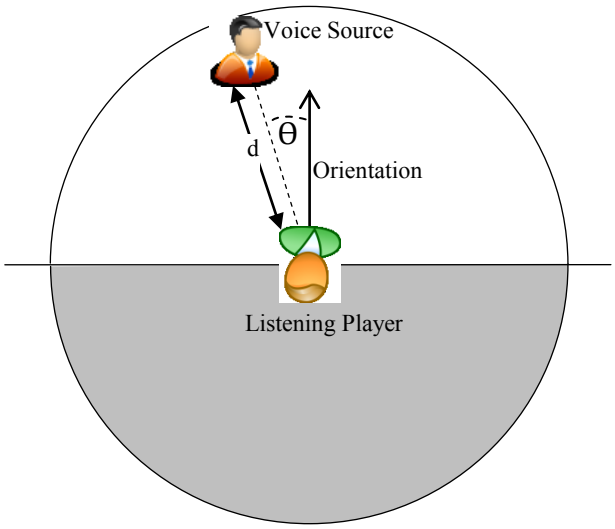

Fig. 4. Illustration of calculating the gain (volume) for each voice source according to the distance and the azimuth angle between the source and the listening player

If the positions of voice sources are behind the listening player (as shown in the dark upper semi-circle of Fig. 5), common stereo speakers are difficult to render such sound effects. Since people are not as sensitive to the sound sources behind them as the sound sources in front of them in either distance or orientation, we treat the sound behind players as relatively plain and weak environmental background sound that listeners are not much concerned. For the voice from the back side of the listening player, we therefore set both the right loudspeaker volume and the left loudspeaker volume to be identical and equal to the half of the base volume, as shown in Eq. (8), for emulating the experience of a monophonic and weak voice.

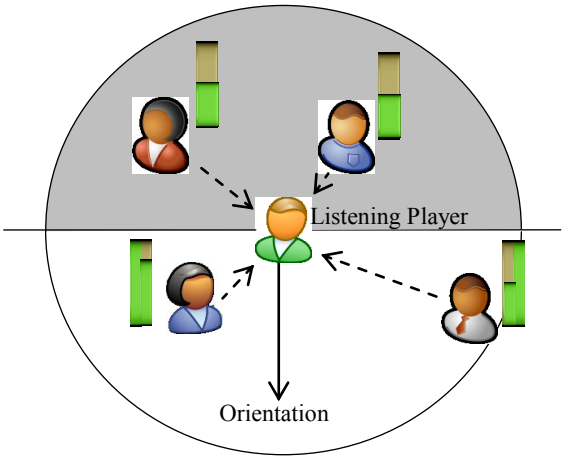

Fig. 5. Illustration of voice balance for different voice sources, where the voice behind the listening player is rendered as monophonic and half-volume

$$Volume_{left} = Volume_{right} = Volume_{base} \times 0.5, \text{ if } 180 \le \theta_{i,j} \le 360 \quad (8)$$

We implement the above-mentioned voice rendering mechanism at the client side with the Microsoft Direct Sound API, which is the most popular tool adopted by MMOGs and PC games. IVC uses a cyclic buffer to process voice data packets received from every voice source, as shown in Fig. 6. The buffer can be used to rearrange the sequence of received voice packets even if they are received in disorder caused by various network transmission delays of different packet paths. The buffer can also be used to overcome the network jitter problem; IVC starts the voice playback only when the amount of voice data received exceeds a playback threshold.

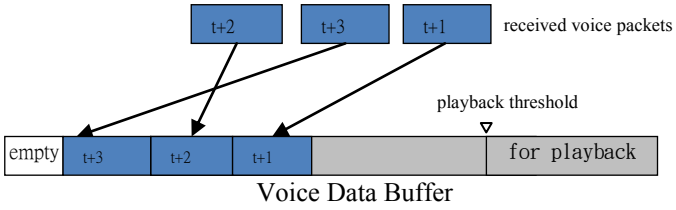

Fig. 6. Illustration of voice data buffer

In addition, we also apply VAST [14] to handle the tasks of mobility management of players, AOI establishment, and (voice) data subscribing and publishing over P2P networking. VAST is a light-weight network library that supports Spatial Publish Subscribe (SPS) so that virtual worlds such as MMOGs can be built with high scalability. Fig. 7 shows the architecture of IVC using VAST to handle the related tasks over P2P networking.

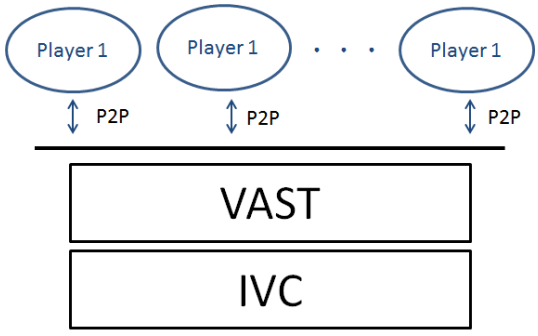

Fig.7. The architecture of IVC using VAST to handle tasks over P2P networking

5 Networking Performance Evaluation

We evaluate the performance of IVC by simulations and compare the simulation results with those of related schemes, namely QuadCast, SectorCast, and NimbusCast [18], to show the advantages of IVC, where NimbusCast stands for the scheme in which the speaking player directly delivers voice data to all its AIO neighbors. The simulation settings are

shown in Table 2. According to [7], we adopt G.728 (16k) and G.723.1 (6.3k) as the standards to encode the listening voices and overhearing voices, respectively. Both G.728 and G.723.1 set the interval between voice payloads to be 30ms; therefore we set the process delay at each node to be 30ms. In the initial phase of a simulation experiment, every player is located at a random coordinate on the virtual world plane, and the player then moves at a speed of 2 units per step with random directions. To avoid the players from quickly swinging between the inside and outside of another player's AOI, we add a buffer zone surrounding the AOI. A player is considered as departing from an AOI only when it moves outside the buffer zone, and a player is considered as entering an AOI only when it really moves inside the AOI.

Table 2. Simulation parameters

Virtual World Size	1000 x 1000 units
The Number of Players	200, 400, 600, 800, 1000
AOI Radius	50 units
Speaking Probability	40%
Listening Voice Data Bandwidth	31.5kbps (G.728(16K))
Overhearing Voice Data Bandwidth	21.9kbps (G.723.1(6.3k))
Voice Data Processing Delay per Hop	30 ms

We perform simulation by using the MIT King data set [11] as the reference of the network latency between arbitrary peers. The data set contains measurements of the latencies between peers in a set of 1,740 DNS servers collected by the King method [17]. We also assume peers have the upload bandwidth distribution proposed in [3], which is shown in Table 3.

Table 3. Upload bandwidth distribution of peers

Uplink (KB/sec)	Fraction of peers
10	0.05
30	0.45
100	0.40
625	0.10

The average latency and the voice data packet dropping rate are the two most important performance metrics. The paper [16] pointed out that a latency of 400 ms is the threshold that people can endure in voice communication. Therefore, incoming voice data packets are dropped by a peer when it has too many voice data packet in the data queue for forwarding. The lower the voice data packet dropping rate is, the higher the quality of voice communication is.

IVC classifies voices into listening voices and overhearing voices, while the other three schemes do not. For the sake of comparison fairness, we assume that the other three schemes can also differentiate the two classes of voices and take advantage of the differentiation to deliver voice data efficiently.

As shown in Fig. 8, the average latencies of all schemes in delivering listening voice data are almost equal. Recall that a peer can only be a listener of another peer. So, the number of listeners equals to the number of peers in the MMOG virtual world, and the traffic of listening voice data is thus not too heavy. Since the listener voice data are sent to the listener with the highest priority, almost every peer has sufficient capacity to send the listener voice data to the listener directly. These explain why there is no distinctive difference among these schemes in latencies of delivering listening voice data. However, as shown in Fig. 9, distinctive differences do exist among these schemes in latencies of delivering overhearing voice data. NimbusCast delivers voice data from the voice source to all peers directly, while other schemes use intermediate peers to forward voice data. Hence, NimbusCast has the lowest average latency among all schemes. As to IVC, its average latency in delivering overhearing voice is always under the threshold (400 ms) and outperforms QuadCast and SectorCast. This is because IVC is latency-aware; it uses NCS to help peers select overhearers having shorter accumulated latencies to deliver voice data with higher priorities.

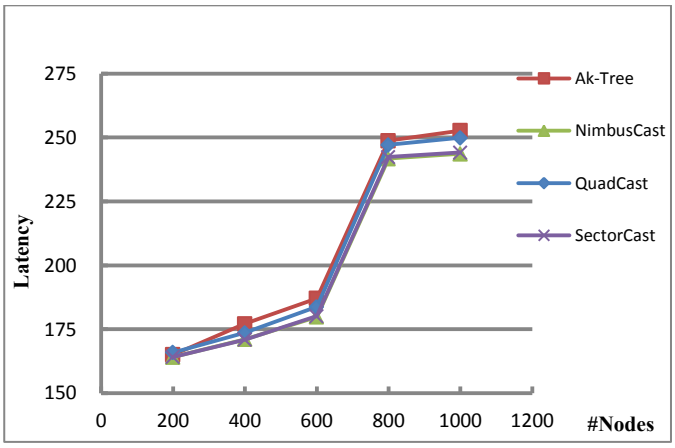

Fig. 8. The average latency in delivering listening voice data

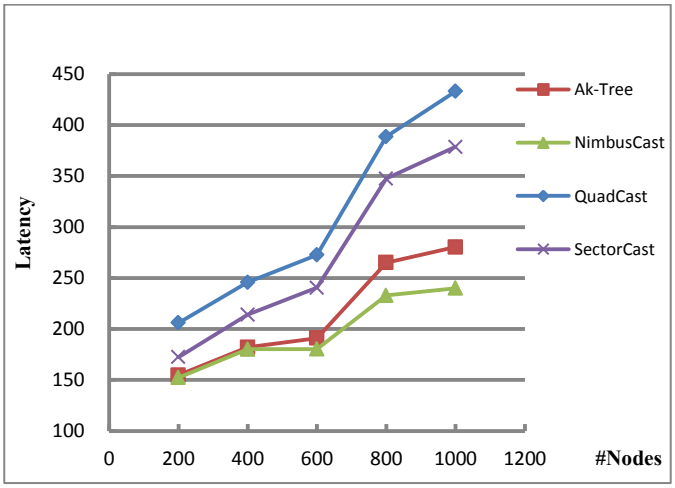

Fig. 9. The average latency in delivering overhearing voice data

Note that all four schemes drop voice data when the peer capacity is insufficient to deliver them, and those dropped data are not considered in calculating the average latency. However, the dropped data affect the quality of voice significantly. Some dropped data even prevent voice being rendered properly. Below, we show the voice data dropping rates of all schemes.

As shown in Fig. 10, NimbusCast, QuadCast and SectorCast have similar dropping rates in delivering listening voice data. This is because the total amount of listening voice data is not very high and peers have the capacity high enough to almost accommodate all listening data delivering. However, the dropping rates of these three schemes in delivering overhearing voice data are quite different, as shown in Fig. 11. This is because the total amount of overhearing voice data is high, peers do not have enough capacity to accommodate all data delivering, and many voice data are thus dropped. SectorCast has higher listening voice data dropping rates than NimbusCast and QuadCast; while QuadCast has evidently higher dropping rates than NimbusCast for the numbers of nodes less than 800.

As shown in Fig. 10 and Fig. 11, IVC is better than the other schemes both in delivering listening voice data and in delivering overhearing data, particularly in the case of smaller numbers of peers. This is because IVC is bandwidth-aware; it uses the adaptive k-ary tree algorithm to dynamically construct data delivering path according to the available capacity of all nodes, which prevents data from being dropped due to insufficient peer capacity.

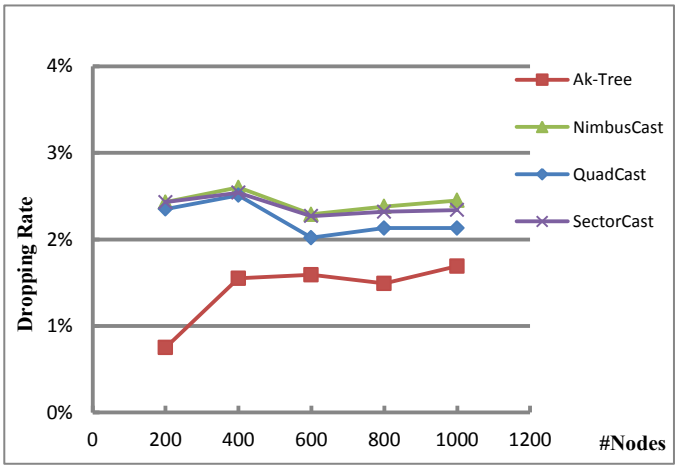

Fig.10. The average dropping rate in delivering listening voice data

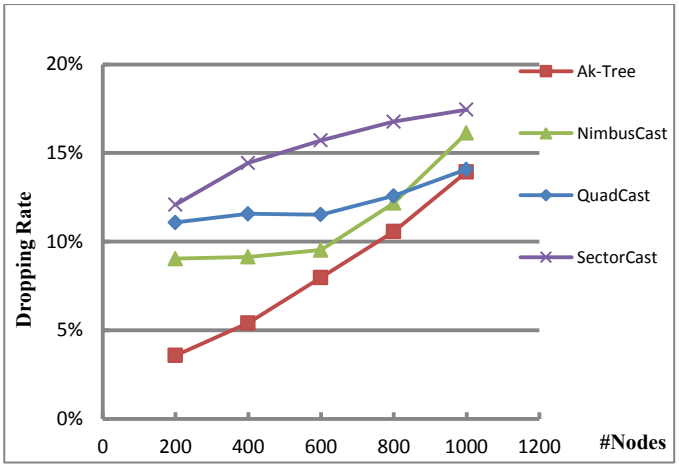

Fig.11. The average dropping rate in delivering overhearing voice data

6 The Analyses Of Users' Experience

The networking performance evaluation shows IVC is an effective bandwidth-aware and latency-aware P2P-based immersive voice communication scheme, as shown in the previous section. In this section, we evaluate whether users feel IVC is good enough in practical usage. We use the Technology Acceptance Model (TAM) [6][30][31] to assess the user experience of using IVC. TAM is an information system theory for explaining the factors users come to accept and use a technology, and for predicting the impact of the use of a technology.

We implement an online virtual gallery guiding system as an instance of the networked virtual environment and integrate IVC into the system. We invite 30 university students to participate in this experiment to act as a **visitor** avatar in the virtual gallery in which there are 8 pictures. One of the paper authors plays the role of the sole narrator to direct visitors to view every picture and to elaborate its background knowledge. Visitors can freely move around the virtual gallery and chat with other visitors and/or narrator.

This experiment has two phases:

1) In the first phase, the guiding system is not equipped with the IVC. Visitors can only hear the plain guiding voice from the narrator but cannot talk with each other. The sound heard by a visitor is static and does not change with what the visitor sees. In other words, the voice communication system is not immersive in the first phase of the experiment.
2) In the second phase, the guiding system is equipped with the IVC. The narrator moves in the virtual gallery picture by picture to introduce every picture. A visitor can hear the voice of the narrator and other visitors if the

visitor can see them in the virtual environment. A visitor can be the listener or the overhearer of a voice source according to its positions, orientation and social relationship. The spatialized sound field rendered with IVC can reveal the relative location of other visitors in the virtual environment and brings participants with good immersive experience.

After these participants complete the experimental operations, we do a questionnaire survey to evaluate the acceptance degree of IVC for users.

The TAM model indicates that when users are presented with a new technology, various factors influence their attitude to accept the technology. Two factors are especially notable: **perceived usefulness (PU)** and **perceived ease-of-use (PEOU)** [6]. According to the TAM model, these two factors will affect the **intention to use** of users, which is the attitude and behavior for using a new technology. We also want to measure the users' experiences of using IVC, so the **awareness** (i.e. perceived features of IVC) and the **appearance** (i.e. the effects of rendering) of IVC are also included in our assessment. Based on TAM, we design a questionnaire containing 20 questions each of which is measured with 7-point Likert scale. The measurement level of each question is quantized into one of the 7 scores in order to be analyzed easily. The seven scores are Strongly Agree (7 points), Agree (6 points), Slightly Agree (5 points), Neutral (4 points), Slightly Disagree (3 points), Disagree (2 points), and Strongly Disagree (1 points).

Table 4. The results of psychological evaluation (n=30)

Grading Items	Average Score
Awareness	5.7
Appearance	6.1
Usefulness	5.8
Easy-of-use	5.2
Intention to use	5.9

Table 4 shows the measurement result of TAM psychological evaluation of IVC. Overall, the average score of every factor is much higher than the neutral score, which means that users take positive attitude towards the use of IVC. In items of awareness (score = 5.7) and appearance (score = 6.1), most participants are aware of the features of IVC and agree that immersive sound effects improve the experience of use for networked virtual environments. Since all participants have completed the two-phase evaluation, they can easily perceive the advancement in presence from plain voice communication to immersive voice communication.

Most participants consider that IVC is helpful for navigating the virtual gallery (score = 5.8), especially for attracting their attention while certain events occur. In our experiments, when the narrator moves to a picture and begins to do a presentation, the visitors can perceive the moving easily via shifted sound fields they have heard. Most participants believe that IVC can promote the playfulness of MMOGs. In the aspect of the PEOU, the score is relatively low (score = 5.2). The following comments from participants may account for the relatively low score. (1) IVC allows players to hear only voices from AOI neighbors, but traditional voice communication mechanisms allow players to hear voices from their team members no matter where the members are located. (2) The voices coming from overhearers have lower quality, which reduces the flexibility for free talking. However, most participants consider that IVC provides an immersive voice communication experience similar to that in the real world. On average, most participants agree that IVC is easy to use without additional efforts.

We got a relatively high score, 5.9, in the "Intention to Use" item. According to the hypotheses of the TAM model, PU and PEOU are the main factors that affect the "Attitude Toward Using", which then affects "Behavioral Intention to Use". Finally, the "Behavioral Intention to Use" affects the "Actual System Use" [6]. We thus observe if our experiment is compliant with these hypotheses by testing the correlation between "Intention to Use" and PU and PEOU. The p-value, viz., the significance level of the correlation test, is set to 0.05 in our evaluation. That is, any p-value less than 0.05 is assumed to confirm the hypotheses compliance. The analysis of this evaluation shows that PU has high positive correlation with "Intention to use" (p-value<0.001); PEOU has positive correlation with "Intention to use" (p-value<0.01). To sum up, our experiment results are consistent with the hypotheses of the TAM theory, which implies that there is adequate validity in this evaluation. The assessment results show that most users think IVC is helpful and easy to use, and they have high intention to use IVC as expected.

7 Conclusion

We have introduced the design of IVC, a peer-to-peer voice communication and rendering scheme, to achieve the goal of providing MMOG players with the immersive experience to hear the voice of neighbor players within AOI without burdening the original MMOG system too many extra

burdens. IVC uses a relation model to classify the voices of AOI neighbor players into the listening voice and the overhearing voice, in which the latter consumes much less bandwidth than the former to reduce the traffic load. IVC also adopts the NCS to reduce the voice transmission latency and constructs the Ak-tree to utilize the complete bandwidth. IVC proposes a spatialized sound rendering method, which adopts general data packets transmission other than connection-oriented streaming technology, to fit the applications in highly dynamic gaming and networking environments.

As shown by the simulation results, the proposed IVC scheme outperforms other related schemes. IVC does not employ complicated in-network mixing scheme and sophisticated spatialized sound rendering techniques, so it is easy to implement IVC on ordinary personal computers or mobile devices. We have implemented IVC and embedded it into an MMOG-like gallery guiding system for evaluating the user experience of IVC under the TAM model. The evaluation results show users regard IVC helpful and easy to use, and thus have high intention to use IVC.

References

1. Altunbasak, A., Apostolopoulos, A., Chou, P. A., Juang, B. H.: Realizing the Vision of Immersive Communication, IEEE Signal Processing Magazine 28(1), 18-19 (2011).
2. Begault., D. R.: 3-D Sound for Virtual Reality and Multimedia, Cambridge, MA: Academic Press Professional (1994)
3. Bharambe, A., Cormac, H., and Venkata N. P.: Analyzing and Improving a BitTorrent Network's Performance Mechanisms, In Proceedings of IEEE INFOCOM (2006).
4. Bharambe, A., Douceur, J. R., Lorch., J. R., Moscibroda, T., Pang, J., Seshan, S., and Zhuang, X.: Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games, In Proceedings of SIGCOMM (2008).
5. Chen, K., Nahrstedt, K.: Effective Location-Guided Tree Construction Algorithms for Small Group Multicast in MANET, In Proceedings of INFOCOM (2002).
6. Davis, F. D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, MIS Quarterly 13(3), 319–340 (1989).
7. Cisco Web Site, http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml.
8. Cowan, N.: The Magical Number 4 in Short-Term Memory: A Reconsideration of Mental Storage Capacity, Behavioral and Brain Science, 87-185 (2001).
9. Dabek, F., Cox, R., Kaashoek, F., and Morris, R.: Vivaldi: A Decentralized Network Coordinate System, In Proceedings of SIGCOMM, August (2004).
10. Gross, M. et. al.: blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence, Proceedings of ACM SIGGRAPH, 819-827 (2003).
11. Krishna, P., Gummadi, S. S., and Gribble, S. D.: King: Estimating Latency Between Arbitrary Internet End Hosts, In Proceedings of SIGCOMM Workshop on Internet Measurment, November (2002).
12. Hu, S. Y., Chang, S. C., and Jiang, J. R.: Voronoi State Management for Peer-to-Peer Massively Multiplayer Online Games, In Proceeding of IEEE Consumer Communication and Networking Conference (CCNC), 1134-1138 (2008).
13. Hu, S. Y., Chen, J. F., and Chen, T. H.: VON: A Scalable Peer-to-Peer Network for Virtual Environments, IEEE Network 20(4), 22-31 (2006).
14. Hu, S. Y., Wu, C., Buyukkaya, E., Chien, C. H., Lin, T. H., Abdallah, M., and Jiang, J. R.: VAST: A Spatial Publish Subscribe Overlay for Massively Multiuser Virtual Environments, VAST Technical Report, March (2010).
15. International Telecommunication Union, ITU-T Recommendation P.59: Artificial Conversational Speech (1993).
16. International Telecommunication Union. ITU-T Recommendation G.114: One-Way Transmission Time (2003).
17. Jiang, J. R., Hung, C. W., and Wu, J. W.: Bandwidth- and Latency-Aware Peer-to-Peer FriendCast for Online Social Network, In Proceedings of P2PNVE (2010).
18. Jiang, J. R., Chen, H. S.: Peer-to-Peer AOI Voice Chatting for Massively Multiplayer Online Games, In Proceeding of ICPADS (2007).
19. Kostadinov, D., Reiss, J. D., and Mladenov, V.: Evaluation of Distance Based Amplitude Panning for Spatial Audio, in Proceedings of ICASSP, 285-288 (2010).
20. League of Legends, http://lol.garena.com
21. Liang, K., Seo, B., Kryczka, A., and Zimmermann, U. R.: IDM: An Indirect Dissemination Mechanism for Spatial Voice Interaction in Networked Virtual Environments, IEEE Transactions on Parallel and Distributed Systems, 24(2), 356-367 (2013).
22. Liang, K., and Zimmermann, U. R.: Cross-tree Adjustment for Spatialized Audio Streaming Over Networked Virtual Environments, in Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video (NOSSDAV '09), 73-78 (2009).
23. Lossius, T., Baltazar, P., and Hogue, T.: DBAP-Distance-based Amplitude Panning, Proceeding of the International Computer Music Conference (ICMC), Montreal, Canada (2009).
24. Naef, M., Staadt, O., and Gross., M.: Spatialized Audio Rendering for Immersive Virtual Environments, Proceedings of the ACM symposium on Virtual reality software and technology, 65-72 (2002).
25. Pulkki, V.: Virtual Source Positioning Using Vector Base Amplitude Panning, J. Audio Eng. Soc. 45(6), 456–466 (1997).

26. Que, Y. P., Boustead, P., and Safaei, F.: Rendering Models for Immersive Voice Communications within Distributed Virtual Environment, IEEE International Region 10 Conference (TENCON), Melbourne, November, 21-24 (2005).
27. Skype, http://www.skype.com/.
28. Teamspeak, http://www.goteamspeak.com/.
29. Triebel, T., Guithier, B., Plotkowiak, T., and Effelsberg, W.: Peer-to-peer Voice Communication for Massively Multiplayer Online Games, In Proceeding of the 6th IEEE Consumer Communication and Networking Conference (CCNC) (2009).
30. Venkatesh, V., Davis, F. D.: A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies, Management Science, 46(2), 186–204 (2009).
31. Venkatesh, V., Bala, H.: Technology Acceptance Model and a Research Agenda on Interventions, Decision Sciences 39(2), 273–315 (2008).
32. Ventrilo, http://www.ventrilo.com/.
33. World of Warcraft, http://us.battle.net/wow/en/.
34. Zimmermann, U. R., and Liang, K.: Spatialized Audio Streaming for Networked Virtual Environments, In Proceeding of the 16th ACM International Conference on Multimedia, 209-308 (2008).