IPSJ Journal Vol. 51 No. 2 517–528 (Feb. 2010)

Regular Paper

A Proposal of an Interactive Broadcasting System for Audience-driven Live TV on the Internet

Yoshia Saito[†1] and Yuko Murayama[†1]

Traditional video services on the Internet are of a broadcasting service nature such as streaming and video-on-demand (VoD). Recent services incorporate more of the interactive nature of network applications such as easy video sharing and those with a chat function. Meanwhile, we have been conducting experimental Internet broadcasting in practice and found it difficult for non-professional broadcasters to provide audiences with satisfactory contents since they do not have a large budget or the technical knowledge to produce broadcasting contents compared to the professional ones. In this paper, we propose an audience-driven broadcast service model in which audiences can send their wish to a broadcaster such that they would like to see some specific objects while broadcasting; the broadcaster can reply back to the request as well. We implemented a prototype system for audience-driven live broadcasting and studied its effects and problems based on the results from the experimental broadcast at our university graduation ceremony and our campus festival. This paper reports our experiments and findings of the audience-driven live broadcasting.

1. Introduction

Broadband Internet services have enhanced video streaming on the Internet to a great extent. People without technical knowledge on broadcasting can contribute and obtain video contents easily through the web-based video-sharing services such as YouTube[1]. Recently, some broadcast services provide users with the functions to communicate with broadcasters and audiences using the interactive nature of the network; Ustream.tv[2] and Stickam[3] are typical ones. Those services provide live broadcast and chat functions such that the broadcaster and its audiences can communicate with each other while watching live streaming. Moreover, personal broadcasting is easy, since one could produce an original live program using widely available software with an economical web camera and a personal computer. The live broadcasting services are being expanded in these days and we presume that small and middle-scale personal broadcasting will become more popular in the future.

In our research, we have been trying and conducting experiments with Internet broadcasting in practice[4]. Through the experiments, we have found that it is quite difficult to provide audiences with quality contents. We are not professional broadcaster with a large budget and technical knowledge. What we need is something we can afford yet we can entertain the audiences.

One of the solutions is to introduce interactivity into the broadcasting. Interactive television (iTV)[5)–8] is a research area which provides interactive features to a television in order to break away from a traditional one-way broadcast service model. iTV provides audiences who watch TV with various interactive functions with shared experiences. Typical iTV systems have chat functions such as text chat and voice chat to let audiences communicate with each other to give them shared experiences[9,10]. They can talk with other audience members who are watching the same TV programme and feel as if they were watching TV at the same place. We believe that audiences would be involved in broadcasting in a more active manner if they could send what they would like to watch to the broadcaster, as if they were members of the broadcaster applying for features of the iTV.

In this paper, we propose an audience-driven broadcast service model, which enables audiences to request the broadcaster to execute some actions, and an interactive broadcasting system called AdlivTV (Audience-Driven LIVe TV system) for the application of the model. In order to explore effects and problems of the audience-driven broadcast as a first step of the research, we implemented a prototype system. The prototype system has several functions for audiences to make a request to the broadcaster. We also study effects and problems of the audience-driven broadcast based on our experimental broadcast at our university graduation ceremony on March 18, 2008 and campus festival on October 25 and 26, 2008. The paper is organized as follows. In the next section, we classify broadcast service models into three groups and clarify what is an audience-driven broadcast service model. Section 3 introduces the model of the AdlivTV system. Section 4 describes related work. Section 5 presents the design of the prototype

†1 Iwate Prefectural University

system of the AdlivTV, its system architecture and user interface. Section 6 reports the experiments using the prototype system and our results. Section 7 gives some conclusions and our future work.

2. Broadcast Service Models

We classify broadcast service models into the following three groups as shown in **Fig. 1** to clarify the service model of our proposal.

2.1 Broadcaster-driven Service Model

The broadcaster-driven service model is the traditional broadcasting service model. In this model, a broadcaster provides contents and the audiences can only do watch them. The communication is one-way and the broadcaster has a lot of power. Most traditional broadcasting services are classified into this category.

2.2 Audience-participatory Service Model

The audience-participatory service model is a recent service model. In this model, a broadcaster can request audiences to execute some actions and the audiences respond to the request. The communication is two-way and the power between the broadcaster and the audience is balanced. Several broadcasting services on the Internet and iTV services are classified into this category. For example, quiz and game show contents [11] on iTV are typical examples of the audience-participatory service model. The broadcaster initiates by giving a question to its audience and receives the answers from them in return. These contents are mainly structured by the broadcaster.

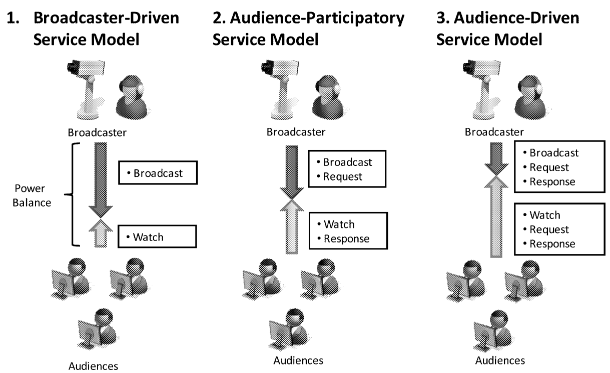

Fig. 1 Broadcast service models.

2.3 Audience-driven Service Model

The audience-driven service model is our proposed service model. In this model, audiences can request a broadcaster to execute some actions and the broadcaster responds to the request. The broadcasting contents are dynamically changed; triggered by the audience's requests. The communication is two-way and interactive communication functions between the audience and the broadcaster are especially enhanced compared with the other service models. The difference between the audience-participatory service model and the audience-driven service model is who initiates the requests. The audience-driven service model enables the audience to send requests to the broadcaster. There are also differences in terms of the request features between the audience-participatory service model and the audience-driven service model. A request within the audience-participatory service model provides a few choices to audience members and it is often a trigger for a branching point of content. A request of the audience-driven service model includes camera work instructions which have parameters of location and orientation of the camera and can change the broadcasting content in real-time.

The feature of this model is that the audience has a lot of power and it complements the audience's demands. Thus, the contents are structured by the audience. On the other hand, the broadcaster does not have much power. The broadcasters, however, are able to provide attractive contents even if they do not have a huge budget or the technical knowledge to produce broadcasting contents to entertain the audience.

We presume that an application of the audience-driven service model is also an application of CSCW (Computer-Supported Cooperative Work) [12] because a broadcaster and its audience cooperate together to create broadcast content. In the audience-driven broadcast, the broadcaster is central to the work and the audience helps the broadcaster's work for its own enjoyment. Technologies of CSCW would be able to be applied to the application of the audience-driven service model.

3. AdlivTV

AdlivTV is an interactive broadcasting system which is based on the audience-driven broadcast service model in terms of realizing audience-initiated requests. It provides several interactive communication functions between audiences and a broadcaster. Allowing audiences to send requests to the broadcaster gives them the power to control broadcasting operations indirectly. The motivation of AdlivTV is to let audiences participate in broadcasts more actively to keep them interested.

Figure 2 shows the system model of AdlivTV. In this model, we define broadcaster as a group of members who work together to broadcast a live content, including server operators and camera operators. Firstly, a broadcaster starts broadcasting a live video streaming to the Internet. Secondly, the audience watches the live video streaming on their computers or TV-like screens. Thirdly the audiences can request some actions to the broadcaster. For example, the requests could be "Look here", "Move here", "Zoom in/out" and so on. The broadcaster waits for the audience member's requests. They send their requests to the broadcaster using input devices such as mice, keyboards, microphones and TV-like controllers. The requests are represented by various ways besides chat. Fourthly when the audience's requests arrives, the broadcaster should respond to the requests as best as possible. The requests and responses are received by all audiences in order to realize a shared experience. Throughout the broadcast, the request and response cycle is repeated.

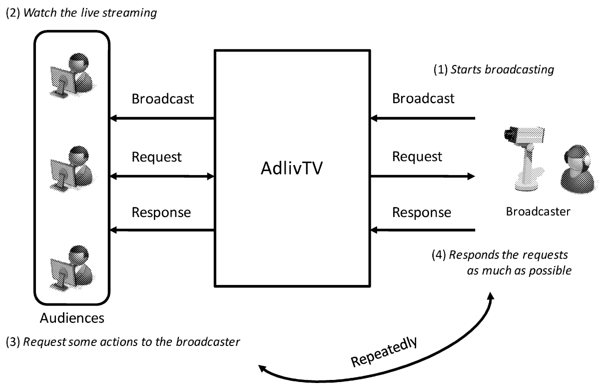

Fig. 2 System model of AdlivTV Model.

The scope of the AdlivTV is small and middle-scale personal broadcasting and the target users are amateur broadcasters with neither any budget nor professional knowledge to produce broadcasting contents. AdlivTV provides the amateur broadcasters with effective broadcasting skills since they can have direct feedback from the audience while broadcasting.

Since AdlivTV enables multiple audience members to communicate with one broadcaster interactively, there is the possibility of competitive requests. Many simultaneous requests from the audience will be an issue for AdlivTV because the broadcaster could not respond to all of these requests in real-time. We assume that AdlivTV does not provide content to mass users and the maximum size of the audiences is several hundred in order to ensure real-time interaction between the broadcaster and the audience. Non-interactive broadcasting can provide content to mass users because they do not need to apply audience feedback in real-time. Several interactive broadcasts such as quiz shows can also provide content to mass users because the variety of responses from the audience is limited and the broadcaster has time to obtain the responses. Since a broadcaster of the AdlivTV must respond to various audience requests in real-time, the number of requests should be restricted so that the broadcaster can address them interactively. Therefore, AdlivTV targets local events such as university graduation ceremony where the size of the audience is several hundred at most.

Since ordinary persons create broadcasting contents with AdlivTV, we assume the broadcasting equipment such as cameras and PCs should be inexpensive. AdlivTV should not require expensive equipment and software to realize personal broadcasting.

4. Related Work

We present several related work on iTV and its interactive functions.

Social TV systems have been proposed by many researchers [13)–15)]. A Social TV system supports audiences' social activity by providing a rich set of communication functions. Audience members can communicate with each other in a community over time and across distances. It lets audiences know what TV programmes their friends and family are watching and makes the audiences aware of the context of their friends and family. To enhance communications, Social TV systems typically provide text chat and voice chat functions for audiences. Geerts[16] studied the difference between text and voice chat systems in terms of an interactive communication function. The paper reports that voice chat enables audiences to continue following the TV programme and younger audiences prefer the text chat.

Coppens, et al.[17] proposed AmigoTV and implemented its prototype system. AmigoTV provides not only chat functions but also visual communication functions. In the system, audience member's avatars are shown on the TV screens such that the presence of other audience members are known. The audience members are also able to show emoticons on their TV screens similar to AdlivTV to express and share their feelings.

Shamma, et al.[18] developed a synchronized video player for online videos to provide a shared experience. They added the video player into an instant messenger so that IM users can watch online videos while chatting with each other. Drugge, et al.[19] studied telepresence, which provides an experience to an audience as if everyone is at the same place. Their system offers various interactive functions to the audience for a shared experience. They use a wearable computer with an HMD to provide ambient telepresence and remote interaction. The system also can convert text into voice or vice versa so that a receiver can choose his/her favorite media.

AdlivTV is different from these iTV work in that it aims to provide audiences with shared experiences as if they were members of the broadcaster. The audience would be involved in broadcasting in a more active manner than with telepresence, because they could send what they would like to watch, to the broadcaster.

5. Implementation

We designed a prototype AdlivTV system to conduct a practical experiment. The implementation of the prototype system is based on Adobe Flash so that audiences can watch live video streaming on their web browsers without specific software. In this section, we describe the system architecture and a user interface with interactive request functions.

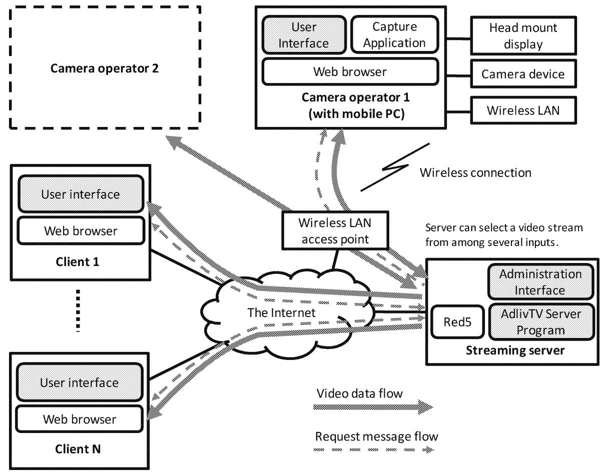

Fig. 3 System architecture.

5.1 System Architecture

The prototype system has a streaming server, several clients and multiple camera operators. **Figure 3** shows the system architecture of the prototype system.

We use Red5 v0.7.0 final to build the streaming server. Red5 is an open source Flash server written in Java and the streaming server has a 3.2 GHz Intel Xeon processor and 2 GB RAM. The streaming server receives video data from the camera operator and broadcasts it to clients. A server program also works on the server to handle requests from clients and manage the system. The administration interface provides a GUI interface for the broadcaster. **Figure 4** shows this interface. (Note: The Japanese version was used and the underlined English parts have been added for the purpose of this explanation.) The max connection function enables the broadcaster to control the number of connections on the server. If undesirable audience members negatively affect the broadcasting, the server operator can ban them using the blacklist function by entering their audience IDs. The streaming server allows a broadcaster to have several video inputs and send one of them to the audience using the switch camera function. The an-

Fig. 4 Administration interface.

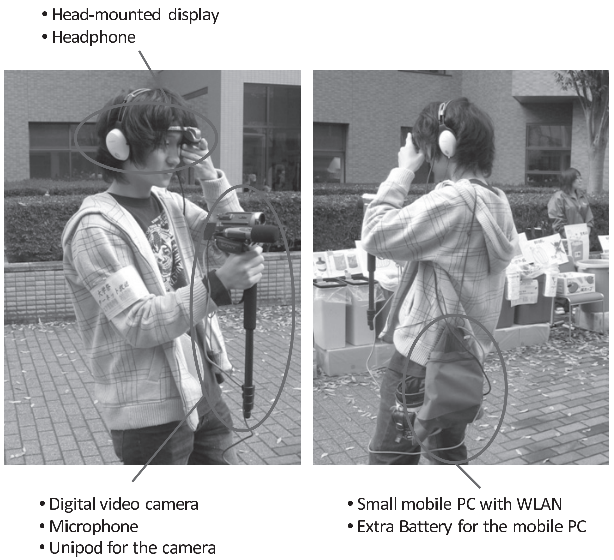

Fig. 5 Camera operator.

nounce function shows a text message on the video display for the broadcaster's announcements. All information on server is shown in the system messages field.

The camera operator has a Head-mounted display (HMD), a camera device and a small mobile PC equipped with wireless LAN (WLAN) as shown in **Fig. 5**. We use the Sony VAIO type-U as the small mobile PC, which has a 1.06 GHz Intel Core 2 Solo processor and 1 GB RAM. The mobile PC and an extra battery are placed in a bag. The camera operator can check the current broadcast and the audience's requests through the HMD. The mobile PC is connected with a WLAN access point and the camera operator can walk around to comply with the audience's various requests. The video data is captured by a capture application implemented in Flash and sent to the streaming server.

When the client accesses to a web page for the broadcast, a Flash application for audiences is launched. At first, the client connects to the AdlivTV server program with a unique number. The server program generates an audience ID using the received number as a seed and sends it to the client. The client keeps the unique number as a cookie on the web browser and can receive the same ID from the server each time. The ID is used for system logs to identify audiences. The logs maintain connection time, disconnection time and the audience requests with their IDs. After obtaining the ID, the client establishes several connections to the server and they can send requests through an original user interface for AdlivTV while watching live video. In the next subsection, we describe this user interface

5.2 User Interface

Both the audience and the broadcaster use the same user interface to watch the live video with the prototype system. **Figure 6** shows the user interface of the AdlivTV (Note: The Japanese version was used and the underlined English parts have been added for the purpose of this explanation.). It is written in ActionScript 3.0 which is a language used to implement Flash applications. The user interface provides a live video player and two functions for the audience members to make requests.

The first function is chat. This popular communication function lets the audience and the broadcaster to exchange text messages with each other. It is often applied in existing interactive TV systems. In AdlivTV, the audiences can send any request to the camera operator using this chat function. Audience members can set their names so that text messages are shown with these names, their IDs and the message times. The camera operator reads the text message and executes the requested action. The chat function is useful for sending complex requests. The broadcaster can also use the chat function to send response messages to the

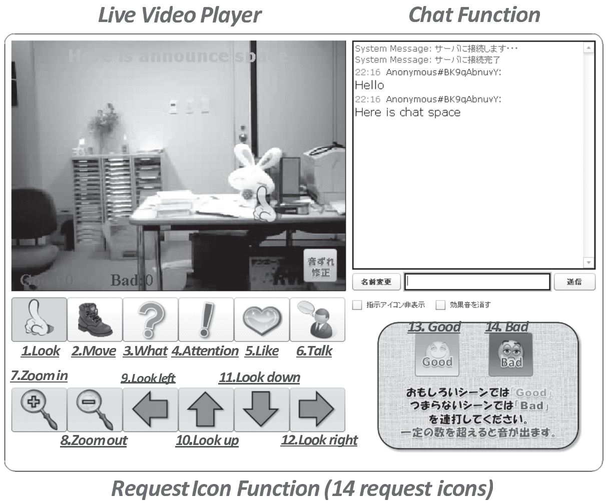

Fig. 6 User interface.

audience. Of course, the audience and the broadcaster are able to communicate with each other freely using the chat function.

The second function is provided by a set of request icon functions. These icons are presented in the menu and the users can select them by clicking them and the request is sent to the broadcaster. We have the following prototype icons; (1) Look: Focus the camera onto the specific object, (2) Move: Move to the specific location, (3) What: Ask about the specific object, (4) Attention: Pay attention to the specific object, (5) Like: Indicate that the specific object is their favorite, (6) Talk: Talk to the specific person, (7, 8) Zoom in/out: Zoom in/out at the current angle, (9, 10, 11, 12) Look left/up/down/right: Turn to the specific direction, (13) Good: Indicate that the current broadcasting is good, (14) Bad: Indicate that the current broadcasting is bad. (Note: The Good and Bad icons increase counters shown at the bottom of the video field.) These icons are unique with our system. The icons provide the audience and the broadcaster with easy and intuitive communication tools. The icons selected by the audience members are reflected in the video in a few seconds. It is useful for stylized requests.

All audience members can use the icon function anytime and the icons are shown in the video field to everyone in the audience. Sometimes several audience members may use the request icon function at the same time and several different objects or locations are highlighted in video field. In this case, the broadcaster (i.e., the camera operator) judges which request should be applied and executes one of them.

6. Experiments

We carried out two practical experiments to study the effectiveness of audience-driven broadcasting and how audiences place requests to the broadcaster. The first experiment was conducted at the graduation ceremony of Iwate Prefectural University on March 18, 2008. The main purpose is to examine the effectiveness of audience-driven broadcasting. We compared audience-driven broadcasting with broadcaster-driven broadcasting to confirm which form of broadcasting is preferred by audiences. The second experiment was conducted at the campus festival of Iwate Prefectural University on October 25 and 26, 2008. The main purpose of this experiment is to study the effects of the implemented request functions and find issues with the system in a practical situation. We examined how audiences use the request functions and their impact.

In each experiment, we distributed a live streaming content to the audience via the Internet. Audience members could watch the content in campus and at home if there is an Internet connection. The audiences included general persons and researchers related to the experiment.

6.1 The Experiment at the Graduation Ceremony

We used an early prototype system for the first experiment at the graduation ceremony. Although the early prototype system had both chat and request icon functions, only a few request icons such as "Look" and "Move" were available. We prepared a Windows Media Server (WMS) for the broadcaster-driven service model to compare with the AdlivTV. We also prepared a top page of the broadcasting where both WMS and AdlivTV pages were linked so that users could select their preferred one.

6.1.1 Environment

Figure 7 shows the network configuration for the graduation ceremony. An AdlivTV server and a WMS were placed in the university. The graduation cer-

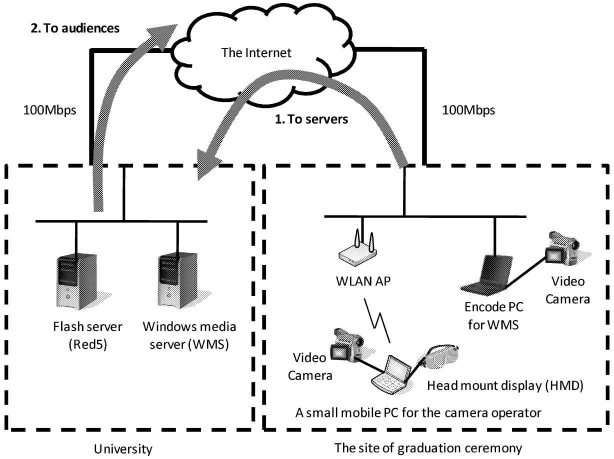

Fig. 7 Network configuration in the graduation ceremony.

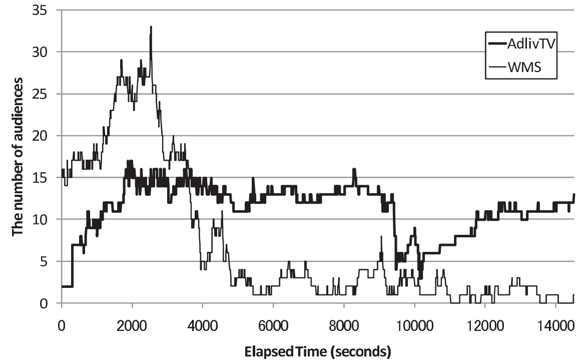

Fig. 8 Changes in the number of audiences every 5 seconds (AdlivTV vs. WMS).

emony was held at other place and we prepared 100 Mbps connections for the venue. For the WMS, an encode PC and a digital video camera were employed. A camera operator for the WMS only used zoom-in-and-out functions and adjusted the camera angle on site. A camera operator for the AdlivTV equipped himself with a small mobile PC, a digital video camera and a HMD, and moved around the hall during the graduation ceremony. Each type of video stream data was sent to the appropriate server located in the university which was then distributed to audiences via unicast. The bit-rate of the video streaming was 200 kbps for both WMS and AdlivTV.

6.1.2 Results

Figure 8 shows the changes in the number of audiences at every 5 seconds with AdlivTV and the WMS. Note that we excluded broadcasters and researchers related to the experiment from these results. The WMS broadcasting had only a few audience members in the middle of the ceremony although it had large audience initially. On the other hand, AdlivTV had fewer audiences than the WMS broadcast at the beginning but kept the audience numbers throughout the middle of the ceremony, while the number of the audiences droppped suddenly at around 10,000 seconds. This was due to a break time and the number of audiences increased again after the ceremony restarted. As a whole, the AdlivTV kept the number of audiences throughout the broadcast while the number of WMS audience members decreased after a short time. The WMS operator used a fixed camera and provided the content with less dynamic camera work. On the other hand, the AdlivTV operator used a mobile camera and provided the content with dynamic camera work with audience-driven functions. The dynamic camera work positively affected the changes in the number of audiences. From this result, we conclude that the audiences prefer AdlivTV over WMS. The result shows that the audience-driven model would support inexpert broadcasters in producing attractive contents.

6.2 The Experiment at the Campus Festival

We used the final prototype system with all of the functions described in Section 4 for this experiment. Since our early prototype system gave favorable results for the first experiment, we evaluated the functions of AdlivTV for two days at the campus festival of Iwate Prefectural University and studied issues to make the system practicable.

6.2.1 Environment

The festival was held at the campus of Iwate Prefectural University. **Figure 9** shows the network configuration for the experiment. Many live performances such as singing and dancing were held on the main stage. A WLAN access point was connected to the AdlivTV server via a LAN in the university. In

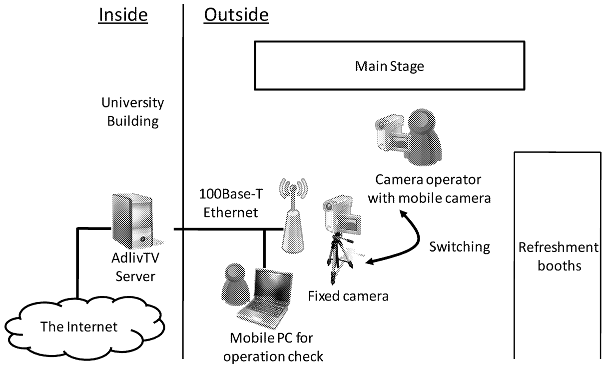

Fig. 9 Network configuration in the campus festival.

Table 1 Total audience.

	25 Oct. 2008	26 Oct. 2008
Total audience	93	200

our experiment, we used two cameras, a mobile one and a fixed one so that the broadcaster could switch the video footage when the camera operator changed the battery of the mobile PC or some other problems occurred. The camera operator walked around the campus with the mobile camera within the coverage of the WLAN access point. The camera operator also moved to the inside of buildings and purchased foods from refreshment booths according to requests from the audience. We used mainly the mobile camera in the experiment.

6.2.2 Results

At first, we counted the total audience size on October 25th, 2008 from 10:00 to 15:00 and on the 26th from 10:00 to 17:00. **Table 1** shows the result. Note that we excluded accesses from broadcasters and fellow researchers from these results. The total size of the audiences was 93 on the first day and 200 on the second day. It was a small-scale broadcast within the scope of AdlivTV and the number of audience members was adequate for our experiment.

We analyzed the changes in the number of audiences for the two days to study the increase and decrease. **Figures 10** and **11** show the changes every 60 seconds. On the first day, an average of around 10 audiences constantly watched the broadcast. On the second day, the average was similar to the 25th, with a maximum of 24 audiences accessing our system before the end of the broadcast because of the grand finale of the festival accompanied by fireworks exploding in the night sky. Many audience members watched the broadcast for the event. In our experiment, we had two types of cameras, viz. the mobile one and the fixed one. We needed to switch often from the mobile one to the fixed one due to various troubles with the mobile one, such as the battery running low in the

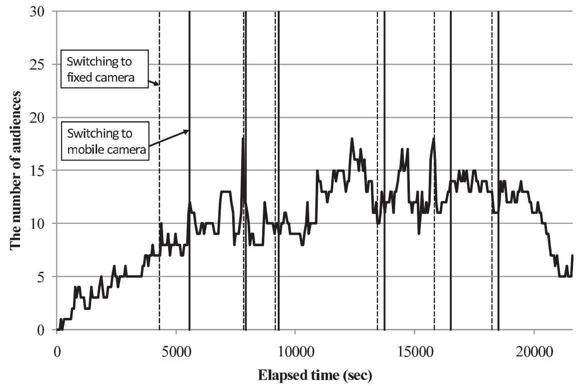

Fig. 10 Changes in the number of audiences every 60 seconds (25 Oct. 2008).

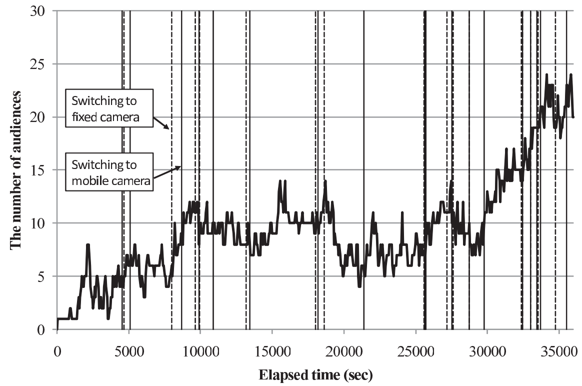

Fig. 11 Changes in the number of audiences every 60 seconds (26 Oct. 2008).

Table 2 Total chat message.

	25 Oct. 2008	26 Oct. 2008
Total chat message (request)	146 (17)	63 (4)

mobile PC attached to the mobile camera. Vertical solid and dotted lines in the graph show the time when we switched to the mobile camera and to the fixed one respectively. We can see that audience rate decreased when we switched to the fixed camera and increased when the mobile camera was used. We presume the reason why the audience size decreased in case of the fixed camera was because we could fulfill requests from the audience such as "Look" and "Move" using the fixed camera. Audience members would feel dissatisfied because these requests were often ignored. From this result, we find that audiences are sensitive to the camera work and broadcasters could get larger audiences if they provided their live programme with audience-driven and dynamic camera work. In this regard, however, it remains possible that only dynamic camera work affected the result and the audience-driven function was not effective. We need to conduct a further experiments to compare the audience-driven model with the case when the broadcaster does not apply audience requests but performs only dynamic camera work. Since the audiences used request icons frequently, as we will see later in this paper, we expect positive effects from our audience-driven will be shown in the experiment.

We also analyzed the use of request functions. **Table 2** shows the result. As for the chat function, the audiences used the chat function 146 times on the first day and 63 times on the second day. Of these chat messages, the number of requests were 17 on the first day and 4 on the second day. Examples of the request message includes "Please go into the specific building", "Move to the best shooting place for fireworks" and "Too much noise, fix it please" and so on. However, the request messages were only ten percent of the total chat messages and the rest were comments on the broadcasting. From this result, we can see that the chat function was not used frequently to send requests to the broadcaster but it was useful for complaints and requests in specific situations.

As for the icon functions, we analyzed the changes in the use of the request icons every 60 seconds and the total use of each request icon. **Figures 12** and **13**

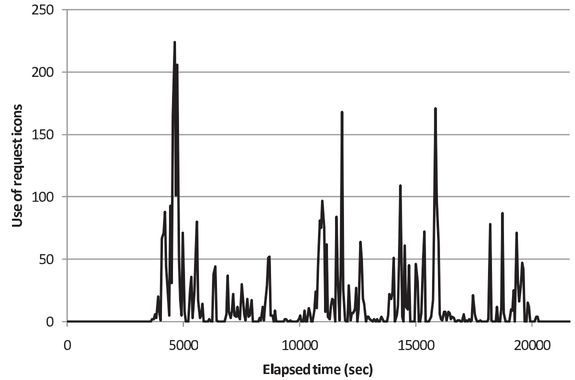

Fig. 12 Changes in the use of the request icons every 60 seconds (25 Oct. 2008).

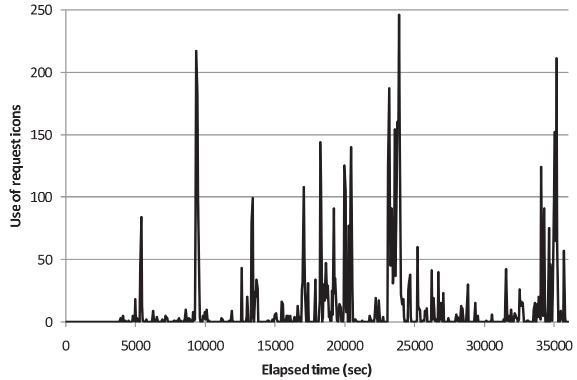

Fig. 13 Changes in the use of the request icons every 60 seconds (26 Oct. 2008).

show the use of the request icons. We can see that the use of the request icons is concentrated in a short period in bursts. We found that several audience members used the request icons at the same time as others. Since the camera operator received conflicting requests should be responded, a burst of requests made him confused. We will need a support system for decision making so that the camera operator can easily select an appropriate one out of hundreds of requests.

Figure 14 shows the total use of each request icon. The frequently-used icons

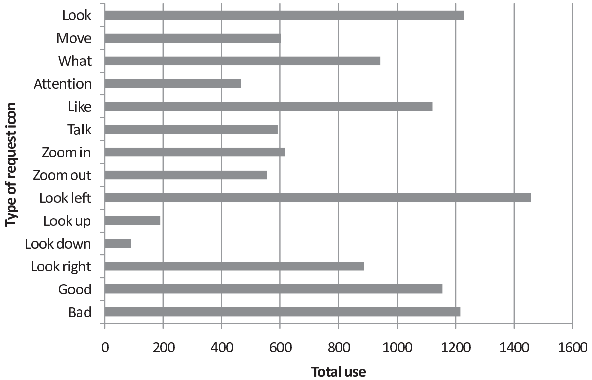

Fig. 14 Total use of each request icon.

include “Look”, “What”, “Like”, “Look left”, “Look right”, “Good” and “Bad”. The “Look” icons would be useful for the audiences to control the camera work. The “What” icon was important to acquire information from the broadcaster and the “Like”, “Good” and “Bad” icons helped to present the audience’s impression to the broadcaster. On the other hand, “Look up” and “Look down” were almost never used in the experience because the main stage was horizontally wide and it did not need vertical camera works. Although the use of “Bad” icon was almost the same as than “Good” icon, it provided opportunities for the broadcaster to assess their camera work and was effective for the inexpert broadcaster. From these results, we found that the audiences could use the request icon functions easily and they were involved actively in the broadcast using these functions. There is, however, the possibility that the new experience of audience-driven broadcasting affects the results. We are considering long-term experiments should be conducted in this regard.

We also found an issue with the request icon functions for the camera operator to respond to the requests. The audiences used the “Talk” icon to a person on the main stage and they continued to use the icon even though the camera operator could not execute the action. Since the camera operator could not indicate that the request was not executable, the audiences kept repeating these useless requests. The audience members were not satisfied when if their requests were ignored many times. The system should provide a response function so that the camera operator can decline inexecutable requests easily. For example, a response icon function which shows a “Decline” icon to the audiences would be useful when the camera operator cannot meet the requests.

6.3 Discussions

In the prototype system, the camera operator decided which requests should be responded to when several requests appeared at the same time. We expected the camera operator could respond to the requests even when several requests appeared at the same time. The experiment results, however, showed that the icons appeared in bursts. The camera operator could handle the requests but experienced significant workload in terms of judgement and the current approach would not be workable when several hundred audiences joined the broadcasting. The workload of the camera operator is a key issue of the current prototype system.

We presume it is difficult to respond to all requests from audience members and we have to drop several requests because of the limits on audience-driven broadcasting. Only one request should appear on the display in any given period of time for the camera operator.

One solution to this issue is to provide a token to an audience member which permits them to use the request icon functions. Although this method is often used to avoid competition and would be effective if there are only a few audience members, they would not be able to wait for the token when there are several hundred audiences.

Another solution is to select and show each request during fixed periods to the camera operator. It will decrease the workload of the camera operator even if several hundred audience members are watching the broadcast. However, audience members whose requests are not selected would feel dissatisfied. The requests should be selected in a fair manner. If the audience members have objection to each other’s requests, it becomes a more complex issue. The audience members whose of which the objections are ignored would feel strongly dissatisfied. We need to treat these requests more carefully and analyze the audience’s behavior to maintain the overall satisfaction. As future work, we will invent several schemes for selecting an appropriate request out of hundreds of requests and clarify which scheme will be most effective for AdlivTV.

Moreover, the system should return the reason for denying their requests to improve their understanding. We expect that this would minimize their dissatisfaction. This function will also apply the decline function as mentioned in Section 6.2. Thus, we should implement not only a support function which enables the camera operator to select an appropriate request but also a function to address audience's dissatisfaction from their requests being declined. We will also implement the feedback function together with the request selection functions.

In the experiment, the camera operator watched the GUI interface which was the same one for audiences and confirmed their requests through an HMD. However, the font size of chat messages was too small for the HMD and it is difficult to confirm the requests when a lot of icons appeared because of the size of the display. In such cases, others associated with the broadcast advised the camera operator using transceivers. Although this method was effectively applied in the experiment, it would not be available if these additional staff were unavailable. Therefore, we should develop a specific user interface for the camera operator.

7. Conclusion

We proposed an audience-driven broadcast service model and an interactive broadcasting system, AdlivTV for the application of the model. We reported our implementation of a prototype system and its evaluation by practical experiments at our university graduation ceremony and campus festival. From the experiments, we have five findings; 1) audiences prefer audience-driven broadcasting than broadcaster-driven broadcasting in small and middle-scale personal broadcasting situations, 2) broadcasters could have more audiences if they provided an audience-driven system combined with dynamic camera work, 3) the icons for requests were used in a short period of time and in bursts, 4) the audiences used the icon function easily and were actively-involved in the broadcast by the use of the function, 5) a response function should be provided for the camera operator to decline difficult requests.

As future work, we will study functions for supporting camera operators since the workload of the camera operator is intensive under the current prototype system. They need to be able to select an appropriate request out of hundreds and decline other requests easily. We will also conduct experiments at various events to evaluate AdlivTV further and to analyze the experimental results as time series data.

References

1) YouTube. http://www.youtube.com/
2) Ustream.tv. http://www.ustream.tv/
3) Stickam. http://www.stickam.com/
4) Takamizawa, N. and Murayama, Y.: Use of Telepresence in Informal Broadcasting over the Internet, *Proc. ACM SIGMM 2004 Workshop on Effective Telepresence*, pp.12–15 (2004).
5) Jensen, J.: Interactive Television – A Brief Media History, *Proc. EuroITV2008*, pp.1–10 (2008).
6) Cesar, P. and Chorianopoulos, K.: Interactivity and user participation in the television lifecycle: Creating, sharing, and controlling content, *Proc. UXTV2008*, pp.125–128 (2008).
7) Ursu, M., et al.: Interactive TV narratives: Opportunities, progress, and challenges, *ACM TOMCCAP*, Vol.4, Issue 4 (2008).
8) Jensen, J.: Interactive television: New genres, new format, new content, *Proc. 2nd Australasian Conference on Interactive Entertainment*, pp.89–96 (2005).
9) Metcalf, C., et al.: Examining presence and lightweight messaging in a social television experience, *ACM TOMCCAP*, Vol.4, Issue 4 (2008).
10) Weisz, H., Kiesler, S., Zhang, H., Ren, Y., Kraut, R. and Konstan, J.: Watching Together: Integrating Text Chat with Video, *Proc. CHI'07*, pp.878–886 (2007).
11) Tuomi, P.: SMS-based human-hosted interactive tv in Finland, *UXTV '08*, pp.67–70 (2008).
12) Reinhar, W., Schweitzer, J., Volksen, G. and Weber, M.: CSCW tools: concepts and architectures, *Computer*, Vol.27, No.5, pp.28–36 (1994).
13) Harboe, C., Metcalf, C., Bentley, F., Tullio, J., Massey, N. and Romano, G.: Ambient Social TV: Drawing People into a Shared Experience, *Proc. CHI'08*, pp.1–10 (2008).
14) Cesar, P., Chorianopoulos, K. and Jensen, J.: Social television and user interaction, *Computers in Entertainment* (CIE), Vol.6, Issue 1 (2008).
15) Oehlberg, L., Ducheneaut, N., Thornton, J., Moore, R. and Nickell, E.: Social TV: Designing for Distributed, Sociable Television Viewing, *International journal of human computer interaction*, Vol.24, No.2, pp.136–154 (2008).
16) Geerts, D.: Comparing Voice Chat and Text Chat in a Communication Tool for Interactive Television, *Proc. NordiCHI'06*, pp.461–464 (2006).
17) Coppens, T., Trappeniers, L., Godon, M.: AmigoTV: Towards a social TV experience, *Proc. EuroITV2004* (2004).

18) Shamma, D., Bastea-Forte, M., Joubert, N. and Liu, Y.: Enhancing online personal connections through the synchronized sharing of online video, *CHI'08 Extended Abstracts*, pp.2931–2936 (2008).
19) Drugge, M., Nilsson, M., Parviainen, R. and Parnes, P.: Experiences of Using Wearable Computers for Ambient Telepresence and Remote Interaction, *Proc. SIGMM 2004 Workshop on Effective Telepresence*, pp.2–11 (2004).

(Received May 18, 2009)
(Accepted October 2, 2009)

(Original version of this article can be found in the Journal of Information Processing Vol.18, pp.26–37.)

Yoshia Saito received his Ph.D. degree from Shizuoka University, Japan, in 2006. He had been an expert researcher of National Institute of Information and Communications Technology (NICT) from 2004 to 2007, Yokosuka, Japan. He is currently a lecturer at Iwate Prefectural University since October 2007. His research interests include internet broadcasting and mobile computing. He is a member of IEICE, IPSJ, IEEE, and ACM.

Yuko Murayama is a Professor at Iwate Prefectural University. She received M.Sc. and Ph.D. both from University of London in 1984 and 1992 respectively. She had been a visiting lecturer from 1992 to 1994 at Keio University, a lecturer at Hiroshima City University from 1994 to 1998. She has been with Iwate Prefectural University since April 1998. Her interests include internetworking, network security and trust. She is a member of IEEE, ACM, IPSJ, IEICE, and ITE. She had been a Director of IPSJ from 2006 to 2007. She serves as the Working Group Coordinator for IFIP TC11.