MULTICAST OF REAL-TIME MULTI-VIEW VIDEO

Li Zuo *, *Jian Guang Lou* †, *Hua Cai* †, *Jiang Li* †

* Dept. of Computer Science, Xi'an Jiaotong University, Xi'an, China
† Media Communication Group, Microsoft Research Asia, Beijing, China

ABSTRACT

As a recently emerging service, multi-view video provides a new viewing experience with high degree of freedom. However, due to the huge data amounts transferred, multi-view video's delivery remains a daunting challenge. In this paper, we propose a multi-view video-streaming system based on IP multicast. It can support a large number of users while still keeping a high degree of interactivity and low bandwidth consumption. Based on a careful user study, we have developed two schemes: one is for automatic delivery and the other for on-demand delivery. In automatic delivery, a server periodically multicasts special effect snapshots at a certain time interval. In on-demand delivery, the server delivers the snapshots based on distribution of users' requests. We conducted extensive experiments and user-experience studies to evaluate the proposed system's performance, and found that the system could provide satisfying multi-view video service for users on a large scale.

1. INTRODUCTION

With the rapid development of electronic and computing technology, multi-view video has recently attracted extensive interest due to greatly enhanced viewing experiences. For example, a system called EyeVision [1] was employed to shoot Super-bowl 2001. Other systems, such as Digital Air's Movia [2] and Wurmlin's 3D Video Recorder [3] were also proposed to capture multi-view video. Later, we proposed an interactive multi-view video system (IMV System) for serving real-time interactive multi-view video service [4]. Unlike conventional single-view video systems, a multi-view video system allows the audience to change view direction and to enjoy some special visual effects such as View Switch and Frozen-moment. It largely enhances user experience in interactive and entertainment orientated applications.

As a recent emerging service, multi-view video provides a new viewing experience with high degree of freedom. However, it also brings challenges to data delivery due to huge data amounts to be transmitted. Hence, an interactive unicast solution was adopted by the previous IMV system in order to support a high degree of interactivity. However, unicast cannot meet the requirements of increasing number of users due to restricted network bandwidth and limited server-processing capability. Different from the conventional unicast streaming, IP multicast is a promising technology that can handle users on a large scale. Many researchers have been investigating this area in the last decade. Among efforts are work in VoD systems [5]. Cooperating with some delivery policies such as command batching [6, 7] and video patching [8, 9], VoD multicast systems can provide users near VoD service and keep relatively low bandwidth costs.

When IP multicast technology is used for implementing a multi-view video delivery system, interactivity becomes a very important issue since multi-view video has unique features. In this paper, our proposed a multi-view video multicast system to support a large number of users and a high degree of interactivity. Based on a detailed user study, we developed two schemes, one for automatic delivery and the other for on-demand delivery. In the automatic delivery, the server periodically multicasts special effect snapshots at a certain time interval. And in the on-demand delivery, the server delivers the snapshots based on distribution of user requests. The proposed system was also evaluated by extensive experiments and user-experience studies.

The rest of the paper is organized as follows. In Section 2, we outline the overall structure of multicast IMV system, and then we present the video delivery schemes of our conventional and special effect videos in Section 2.2. Some experimental results are presented and discussed in Section 3. In Section 4, we conclude our work.

2. VIDEO DELIVERY SCHEME

2.1. System Overview

Our IMV system mainly consists of three parts [4]: capturing part, server part and client part. The capturing part captures a dynamic event simultaneously from multiple cameras with various view directions. The captured video signals are compressed in several control PCs and then sent to the server through a network backbone, e.g., a gigabit Ethernet. The server part collects both the N compressed video streams from the control PCs and transcoded special effect snapshots from the transcoding servers. It then provides a multi-view video service to end users.

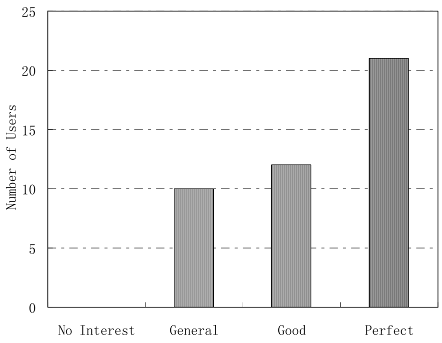
Interactive special effects such as frozen moment and view sweeping, and view switching are three important features of our IMV system. In the frozen moment, time is frozen and the camera view direction rotates about a given point, while view sweeping involves sweeping through adjacent view directions while time is still moving (please refer to figure 2 of [4] for more description). View switching means that users are able to switch from one camera view direction to another as the video continues along time. Through a usability study, we found that users were really interested in these new features. In the study, more than 40 people are invited as participants, including people with technical and non-technical background. The results are summarized in Fig.1. Fig.1 (a) indicates that about 75% of the participants consider view switching is a very useful feature in an IMV system. Meanwhile, in Fig.1 (b) about 90% of them consider frozen moment is a very interesting feature.

Based on these observations, we mainly focused on how to provide the exciting multi-view video features for users based on IP multicast techniques.

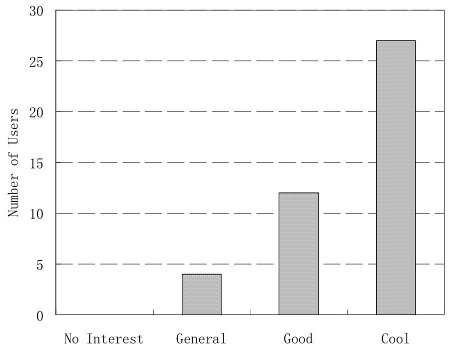
---

[*] The work presented in this paper was carried out in Microsoft Research Asia.

1-4244-0367-7/06/$20.00 ©2006 IEEE



(a)User study result on view switching.



(b) User study result on frozen moment.

Fig. 1. User study results on view switching and frozen moment.

2.2. Multicast Video Delivery

In general, the server should broadcast videos of conventional views and special visual effects. However, such a delivery scheme needs huge network bandwidth, especially when the number of viewpoints becomes large. To handle this problem, we multicast the video contents through $M + N$ video channels. Here the M channels are assigned to multicasting the videos from different conventional views, while the N channels are used for delivering the special effect streams. As shown in Fig.2, each client simultaneously joins one conventional video channel and one or more special effect channels. The number of the special effect channels that a user joins depends on available downlink bandwidth. Therefore, users with higher available downlink bandwidth can join more special effect channels, and thus can enjoy the special effects with higher degree of interactivity.

2.2.1. Conventional Video Channels

One problem of designing the proposed delivery system was how to select views for the M conventional view channels. In real world scenarios, we found that the number of conventional view channels M is not necessarily the same as the number of viewpoints. Because of the small visual difference between adjacent views, users are unlikely to do a switch operation between the two close viewpoints. In our experiments, we found that $M = 6$ can usually meet user requirements for a total capture angle of 90°.

In our system, view switching is realized by switching from the source to the destination conventional channel. The maximum latency of the view switching is $T_s + T_v$, where T_s is the time of network channel switching and T_v is the latency from the current frame to the next I frame. The value of T_v is determined by the group of picture (GOP) size when compressing the conventional view videos. In our system, the GOP size is set to 1 second. Thus the maximum value of T_v is 1 second and the average value is 0.5 second.

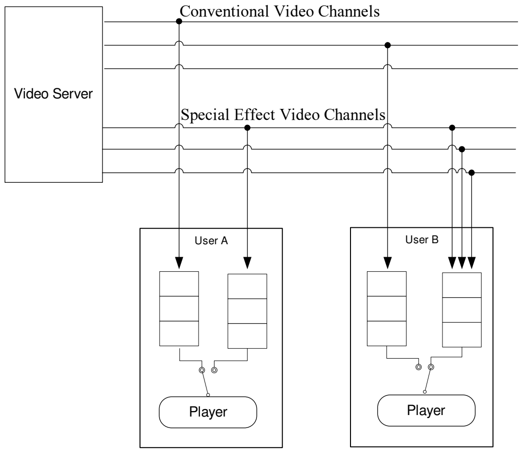


Fig. 2. The overview of IP multicast for online user service.

2.2.2. Special Effect Channels

Due to the limited downlink bandwidth, users cannot get all the special effect snapshots in real time. Fortunately, through the user study in [4], we found that different users often have the similar judgment about exciting moments in a multi-view video, and they will subscribe to special effects when there is an exciting moment. This means that not all snapshots need to be sent to end users. Then the problem is that, for a given available downlink bandwidth, how to select proper special effect snapshots for end users?

For an off-line multi-view video, suppose that the distribution of all user subscriptions $f(t)$ on special effect snapshots is known beforehand. Then, we can find the optimal snapshots p_i $(i = 1 \dots n)$ by minimizing the total differences from the snapshot that a user wants.

$$\arg\min_{p_i}\left(\sum_{i=1}^{n}\int_{\phi_i}^{\phi_{i-1}}(t-p_i)^2 f(t)dt\right) \tag{1}$$

Note that Eq.1 is very similar to the classic scalar quantization problem. The iterative method proposed by Lloyd [10] can be used to estimate the optimal values of p_i. However, in on line multi-view streaming, the distribution function $f(t)$ cannot be known in advance. In other words, we are not able to determine the proper snapshots beforehand.

A very simple strategy, named as automatic delivery scheme, is that the special effect channels multicast the snapshots with a fixed time interval d_s ($d_s \geq T, T = b/B$ is the minimal time interval of sending a snapshot that is determined by the average snapshot size b and the available bandwidth B). In the automatic delivery, all of the sent snapshots are equally distributed in the special effect channels. Obviously, the disadvantage is that the sent snapshots may not be the ones that most users subscribe, because there is no interactivity between users and the server.

To overcome the problem in the automatic delivery, we also design an on-demand delivery scheme that takes user subscriptions into consideration. In the on-demand delivery, the server collects user requests and fetches an appropriate snapshot for most users. The snapshot will be sent when both of the following formulas are satisfied.

$$\begin{cases} C \geq \tau \times S \\ T_\eta \geq T \end{cases} \tag{2}$$

where T_η is the time interval between the sent time of the two snapshots, C is the sum of user requests in the period of T_η, τ is a threshold (0 $\sim 100\%$), S is the total number of logon users, and $T = b/B$.

To better illustrate the process of our on-demand delivery scheme, we give an example in Fig 3. The curve is the distribution of user requests. t_0 is the sent time of the last snapshot, t_s is the snapshot ordered by a user request x, t_1 is the sent time of the current snapshot, and t_c is a proper snapshot for most users. In the on-demand delivery, the special effect video service tries to meet the interaction requirements of most users. It can dynamically adjusts the sending frequency based on the number of user requests and the threshold τ. Therefore, more bandwidth cost can be saved when there are fewer requests. However, the disadvantage is that it brings extra interaction latency for the server needs to collect user requests.

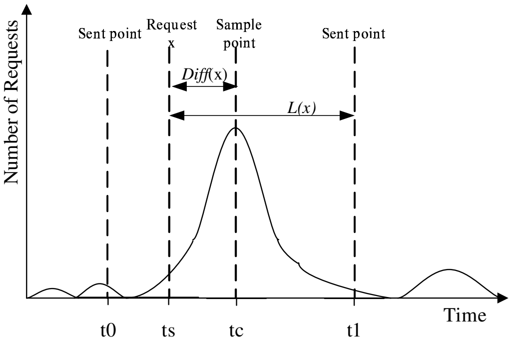


Fig. 3. On-demand delivery scheme. Here, t_0 is the sent time of the last snapshot, t_s is the snapshot ordered by a user request x, t_c is the snapshot of most users, t_1 is the sent time of the current snapshot.

In order to demonstrate the performance and features of the system, we carried out a lot of experiments and user-experience studies. The results can help us select a proper video streaming strategy.

3. EXPERIMENTS

In this section, we describe the experiments on the performances of the automatic delivery scheme and the on-demand delivery scheme under various network conditions.

3.1. Performance Metrics

Before the experiments, we first figured out two metrics that can be used to evaluate the system performance. Here are the definitions:

Special Effect Latency $\mathbf{D(x)}$ Special effect video latency is the time interval from the moment that users send out requests to the moment that the special video starts to be played. In Fig.3, $L(x)$ is the time interval from the request time t_s to the sent time t_1. If t_n is the network RTT, the latency of the command should be $xD(x) = L(x) + 2 \times t_n$, because a user sends command x at $t_s - t_n$, while receives the response at $t_1 + t_n$.

Special Effect Difference Diff(x) Special effect video difference is the time difference between the effect snapshot the sever sends out and the one requested by a user. For example, in Fig.3, *Diff*(x) is the video difference of the command x from the time stamp t_s of the snapshot that a user requests to the time stamp t_c of the snapshot that the server sends out.

3.2. User-experience Study

Even given the values of latency and difference, we still have no clear knowledge about whether they can meet most user requirements. Therefore, it is necessary to study user experiences on various values of latency and video difference. In this paper, we conducted a user experience study, which is formed from the feedback from 43 users after they observed the videos (including Chinese martial arts and gymnastics) with different latency and difference values. The result is shown in Fig.4, where the height of a bar represents the number of users who consider the interactivity with corresponding latency and difference as acceptable.

From Fig.4, we find out that more than 90% of users consider the performance of special effects video as very good when the difference and latency are set to 0.3 seconds. Less than 15% of participants can tolerate the 0.7 seconds latency and video differences. And the configuration of 0.5 seconds latency and 0.3 seconds difference is also acceptable. Most users felt that the latencies and differences in Chinese Martial Art videos are not as comfortable as they are in the gymnastics videos. This means that user responses to different video contents are slightly different. Based on the results, we find out that for a practical system, the latency and difference should be less than 0.5 and 0.3 seconds.

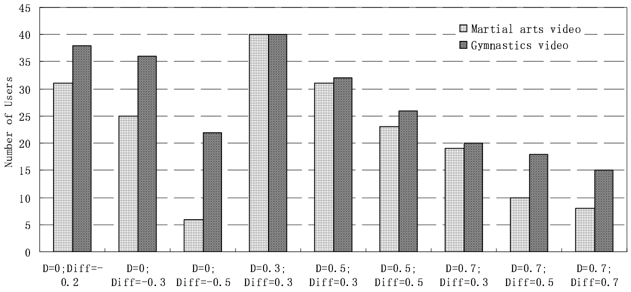


Fig. 4. User study on latency and difference.

3.3. Experiments on special effects video delivery

Fig.5 shows the results of two delivery schemes with different available bandwidth. The experiments were carried out in a LAN with bandwidth of 100Mbps. The run trip time (RTT) is less than 10 ms, and can be neglected in our experiments. In Fig.5(a) and (b), two straight dotted lines are the average values of $D(x)$ and *Diff*(x) in the automatic delivery scheme, while the two curves are the average values of $D(x)$ (the curve with triangle points) and *Diff*(x) (the curve with quadrate points) as the threshold increases in the on-demand delivery scheme. Fig. 5 (c) and (d) are the corresponding average bandwidth costs of the two schemes. As shown in Fig.5 (a), the latency and difference of the automatic delivery and the on-demand delivery are very close when we set a very small threshold (e.g., $\tau = 5\%$). Meanwhile, Fig.5 (b) shows that the on-demand delivery scheme ($\tau < 12\%$) will have a smaller difference than the automatic one when the bandwidth is relatively small (e.g., $B < 1.2$ Mbps). The reason is that the sent snapshots are selected to meet the subscriptions of most users in the on-demand delivery. Furthermore, from Fig.5 (c) and (d), we learn that the on-demand delivery scheme can largely reduce the average downlink bandwidth request. This is because that, in the on-demand delivery, snapshots are only sent when there are user requirements in the system. It seems that if a system has large available downlink bandwidth (e.g., $B > 2.4$ Mbps), both schemes are able to meet user requirements, but the automatic scheme is better because the server does not have to manage any user request. On the other hand, the on-demand scheme will be a better choice when the downlink bandwidth is less than 2.4 Mbps, due to lower latency and differences. Finally, we want to point out that, although the results in Fig.5 come from the videos of Chinese Martial Arts, we can draw a similar conclusion from the results of gymnastics videos which are not presented here due to the space limitations.

4. CONCLUSION

In this paper, we propose a multi-view video streaming system based on IP multicast. The multi-view videos are transmitted through $M+N$ video channels. This multiple-channel scheme can support various users who have different available bandwidth. Furthermore, two multicast delivery strategies, automatic delivery and on-demand delivery, are presented and evaluated in this paper. Based on the proposed streaming schemes, our system can serve users on a large scale, and provide satisfying interactivity for most of them. Moreover, the analysis can also facilitate selecting proper streaming strategy for different multi-view video applications.

5. REFERENCES

[1] “Eyevision project,” in *http:// www.ri.cmu.edu /projects /project_449.html*, 2001.

[2] “Digital movia camera systems,” in *http:// www.digitalair.com /techniques*.

[3] S. Wurmlin, E. Lamboray, O. G. Staadt, and M. H. Gross, “3d video recoder,” in *Proc. of Pacific Graphics ’02*. ACM, October 2002, pp. 325–334.

[4] Jian-Guang Lou, Hua Cai, and Jiang Li, “A real time interactive multi-view video system,” in *Proc. of ACM Multimedia*. ACM, November 2005, pp. 161–170.

[5] Huadong Ma and Kang G. Shin, “Multicast video-on-demand services,” in *ACM SIGCOMM*. ACM, 2002, pp. 31–43.

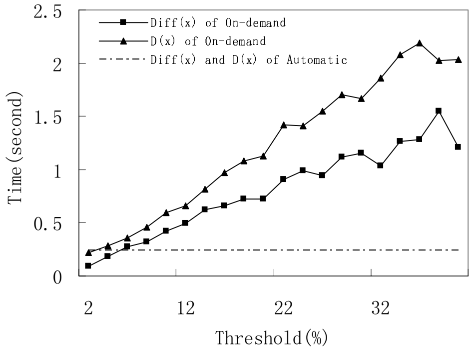
[6] W.F. Poon and K.T. Lo, “New batching policy for providing true video-on-demand (t-vod) in multicast system,” in *Proc. of IEEE International Conference on Communications*. IEEE, 1999, pp. 983–987.

[7] A. Dan, D. Sitaram, and P. Shahabuddin, “New batching policy for providing true video-on-demand (t-vod) in multicast system,” in *Proc. of ACM Multimedia*. ACM, 1994.

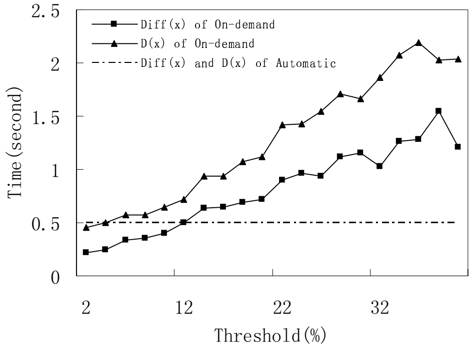
[8] Y. W. Wong and Jack Y. B. Lee, “Recursive patching - an efficient technique for multicast video streaming,” in *Proc. of ICEIS*. ACM, 2003, pp. 306–312.

[9] K. A. Hua, Y. Cai, and S. Sheu, “Patching : a multicast technique for true video-on-demand services,” in *Proc. of ACM Multimedia*. ACM, 1998, pp. 191–200.

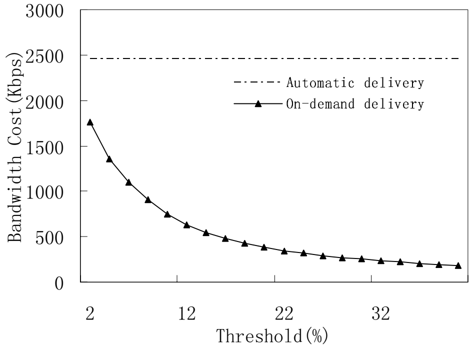
[10] S. P. Lloyd, “Least squares quantization in pcm,” *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.



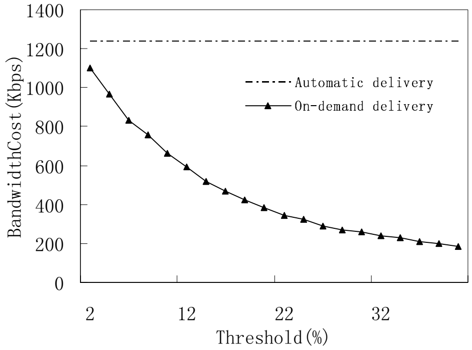
(a) $D(x)$ and *Diff*(x) when $B = 2.4$ Mbps.



(b) $D(x)$ and *Diff*(x) when $B = 1.2$ Mbps.



(c) Average bandwidth cost in Fig.5(a).



(d) Average bandwidth cost in Fig.5(b).

Fig. 5. Average latency and difference vs. thresholds.