PEER-TO-PEER MULTICAST LIVE VIDEO STREAMING WITH INTERACTIVE VIRTUAL PAN/TILT/ZOOM FUNCTIONALITY

Aditya Mavlankar, Jeonghun Noh, Pierpaolo Baccichet, and Bernd Girod

Information Systems Laboratory, Department of Electrical Engineering
Stanford University, Stanford, CA 94305, USA
Email: {maditya, jhnoh, bacci, bgirod}@stanford.edu

ABSTRACT

Video streaming with virtual pan/tilt/zoom functionality allows the viewer to watch arbitrary regions of a high-spatial-resolution scene. In our proposed system, the user controls his region-of-interest (ROI) interactively during the streaming session. The relevant portion of the scene is rendered on his screen immediately. An additional thumbnail overview aids his navigation. We design a peer-to-peer (P2P) multicast live video streaming system to provide the control of interactive region-of-interest (IROI) to large populations of viewers while exploiting the overlap of ROIs for efficient and scalable delivery. Our P2P overlay is altered on-the-fly in a distributed manner with the changing ROIs of the peers. The main challenges for such a system are posed by the stringent latency constraint, the churn in the ROIs of peers and the limited bandwidth at the server hosting the IROI video session. Experimental results with a network simulator indicate that the delivered quality is close to that of an alternative traditional unicast client-server delivery mechanism yet requiring less uplink capacity at the server.

Index Terms— peer-to-peer video streaming, interactive region-of-interest, pan/tilt/zoom

1. INTRODUCTION

Compared to content delivery networks, peer-to-peer (P2P) multicasting is appealing as it requires much less server resources and is self-scaling as the resources of the network increase with the number of users. Recently, numerous academic and commercial Internet P2P video streaming systems have become available, for example [1–4]. However, the interactive features offered by these systems are limited to video-on-demand and/or VCR like functionality.

An early attempt to employ application-layer P2P multicast for live interactive 3DTV is [5]. The P2P overlay in [5] delivers a subset of views to a peer from a set of multiview videos of the scene. It should be noted that for any frame interval, entire views are either selected or dropped according to the peer's viewpoint. Although the authors report latency of interaction, bandwidth saving at the server, etc., they do not report objective metrics like PSNR for the video rendered on the participating peer's display.

In this paper, we propose P2P multicast for delivering video with interactive region-of-interest (IROI) to a population of peers. Different users can watch different regions of the scene with arbitrary spatial resolutions (zoom factors). We build a P2P overlay using a distributed protocol. The goal is to exploit the overlap in the ROIs of the peers by adapting the topology of the overlay according to the changing ROIs. This enables the server to host a live IROI video session with modest uplink capacity and the system scales well with increasing number of peers. Our distributed IROI P2P protocol builds complementary multicast trees for pushing relevant data to the clients and is based on the Stanford P2P Multicast (SPPM) protocol [6, 7].

2. USER INTERFACE AND VIDEO CODING SCHEME

We have developed a graphical user interface which allows the user to select the ROI while watching the video. The ROI location and zoom factor are controlled by operating the mouse. The application supports continuous zoom to provide smooth control of the zoom factor. In addition to the ROI, we also display a thumbnail overview with a rectangular box overlaid to show the location of the ROI.

We require a video coding scheme that efficiently supports random access to arbitrary ROIs, while keeping the transmission rate as low as possible. We have proposed such a video coding scheme in our earlier work [8]. The coded representation of the scene consists of the thumbnail version and versions with different spatial resolutions that are dyadically spaced.

The thumbnail overview, also called as the base layer video, is coded using I, P and B pictures of H.264/AVC. The reconstructed base layer video frames are upsampled by a suitable factor and used as prediction signal for encoding video corresponding to the higher resolution layers. Each frame belonging to a higher resolution layer is coded using a grid of rectangular P slices. Employing only upward prediction enables efficient random access to local regions within any spatial resolution. For a given frame interval, the display of the client is rendered by transmitting the corresponding frame from the base layer and few P slices from exactly one higher resolution layer. We transmit slices from that resolution layer which corresponds closest to the user's current zoom factor. At the client's side, the corresponding ROI from this resolution layer is resampled to correspond to the user's zoom factor. If some enhancement layer P slices are unavailable, we perform error concealment by upsampling portions of the thumbnail video signal.

3. IROI P2P VIDEO MULTICAST

Users participating in the video multicast request different regions of the video at different zoom factors according to individual ROIs. Fig. 1 illustrates an example of the system serving three users. Our IROI P2P streaming system adopts a tree-based approach, similar to [6, 7, 9], for pushing relevant media data to the clients. We build one multicast tree for the base layer and one multicast tree each for every slice of the higher resolution layers, also called enhancement

This work has been supported, in part, by the Max Planck Center for Visual Computing and Communication, Deutsche Telekom Laboratories, and Sharp Laboratories of America.

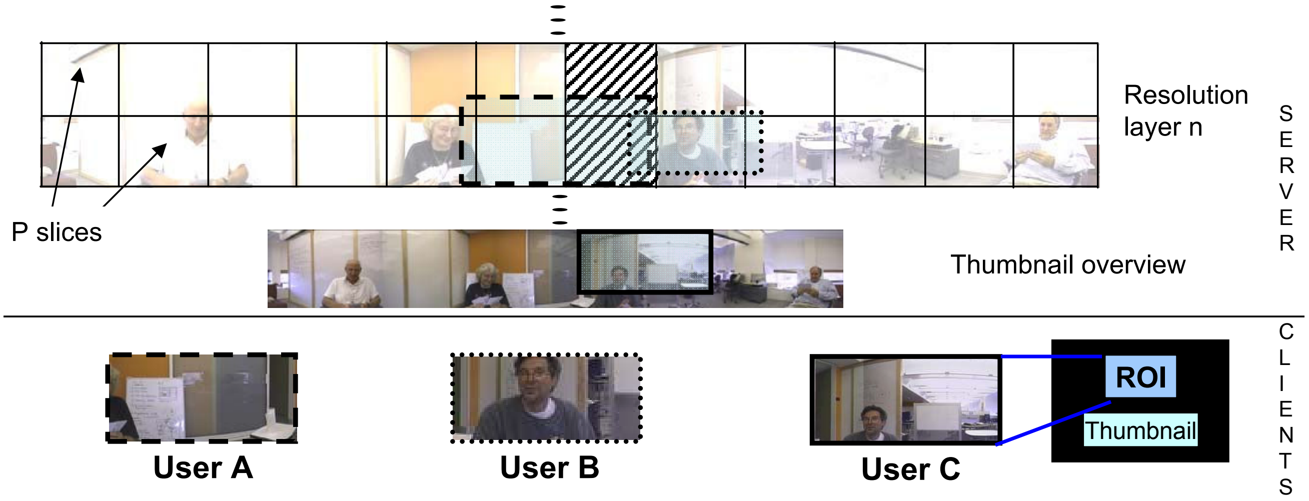

Fig. 1. An ROI is rendered from a portion of the multi-resolution representation. The multi-resolution representation consists of dyadically spaced resolution layers and the base layer. The base layer is displayed as the thumbnail. The ROIs of three users, illustrated within the multi-resolution representation, appear to be of different sizes due to arbitrary non-dyadic zoom factors. The slices shown shaded are required by two users. The video screenshot is taken from the *Cardgame* sequence used for our experiments.

layer slices. Every client subscribes to the base layer tree for the entire session. Depending on its current ROI, each client further subscribes to the trees corresponding to required enhancement layer slices. Clients also dynamically unsubscribe slices that are no longer required for the current ROI. Notice that in case the user's zoom factor corresponds closer to the base layer than any other resolution layer, then the peer needs to subscribe only to the base layer; i.e., the ROI would be rendered using part of the thumbnail, for example user C in Fig. 1.

3.1. Design for Low-Latency of Interaction

At the client's side, a new ROI is rendered immediately upon user input, without waiting for new data to arrive. If the client would delay rendering the requested ROI until new data arrive, the induced latency might hamper the experience of interactivity. In order to render the ROI instantly despite the delay of packets, the client predicts the user's future ROI d frame-intervals (or D seconds) in advance and, if required, initiates connection to new trees beforehand. In our earlier work [10], we have proposed and evaluated several ROI trajectory predictors. The class of video-content-aware predictors performs particularly well and requires that at least d future frames of the base layer are available at the client when the current frame is rendered. To facilitate this, we advance the base layer transmission by S seconds compared to the enhancement layer slices from the source itself. Let $T_{B,n}$ denote the time when frame n of the base layer emanates from the server. For some slice X of some enhancement layer, let this time be denoted by $T_{X,n} = T_{B,n} + S$, where n is the frame index. Finally, a pre-roll delay of C seconds is used to account for delay-jitter in the arrival of the enhancement layer packets; i.e., at time $T_{X,n} + C$, both the thumbnail and the ROI are rendered for frame n.

The parameters of this design are depicted in Fig. 2. For slice X, let X_{e2e} denote the worst-case end-to-end delay, i.e., the longest time it takes for a packet of slice X to reach the client from the server. Notice that $C > X_{e2e}$ helps avoid late-losses. Similarly, for the base layer, let B_{e2e} denote the worst-case end-to-end delay. Irrespective of the value of C, choosing $S \geq D + B_{e2e}$ ensures that d future frames of the base layer are available in the client's buffer at the time of displaying the current frame; i.e., up to frame $n + d$ available at (or before) time $T_{X,n} + C$.

Notice that when frame $n - d$ is displayed at time $T_{X,n-d} + C$, the ROI for frame n is predicted by observing the mouse-moves up to frame $n - d$ and using the buffered thumbnail frames up to frame n. This implies that the join request for new slices for frame n cannot be sent earlier than $T_{X,n}+C-D$ or $T_{X,n-d}+C$. The lookahead d is chosen to be sufficiently large in order to join new trees and receive the required new slices before their display deadline $T_{X,n} + C$.

3.2. Distributed Protocol

We base our protocol on the SPPM protocol [6, 7] for P2P live video streaming. The server maintains a database of slices that each peer is currently subscribed to. Whenever the ROI prediction indicates a change of ROI, the peer sends an ROI-switch request to the server. This consists of the top-left and bottom-right slice IDs of the old ROI as well as the new ROI. In response to the ROI-switch request, the server sends a list of potential parents for every new slice that the peer needs to subscribe. For every slice, we limit the number of peers the server can directly serve, and the server includes itself in the list if this quota is not yet full. The server also immediately updates its database assuming that the peer will be successful in updating its subscriptions. After receiving the list from the server, the peer probes potential parents for every new slice. If it receives a positive reply, it immediately sends an attach request for that slice. If it still fails to connect, then the peer checks for positive replies from other probed peers and tries attaching to one of them. When the ROI prediction indicates a change of ROI, the peer waits a while before sending leave messages to both its parents as well as its children for slices that its ROI no longer intersects. This ensures that slices are not unsubscribed prematurely. The respective parents stop forwarding data to the peer for the respective slices. The respective children request potential parents' lists from the server for the respective slices. In addition, if no data are received for a particular slice for a timeout interval, the peer assumes that the parent is unavailable and tries to rejoin the tree by enquiring about other po-

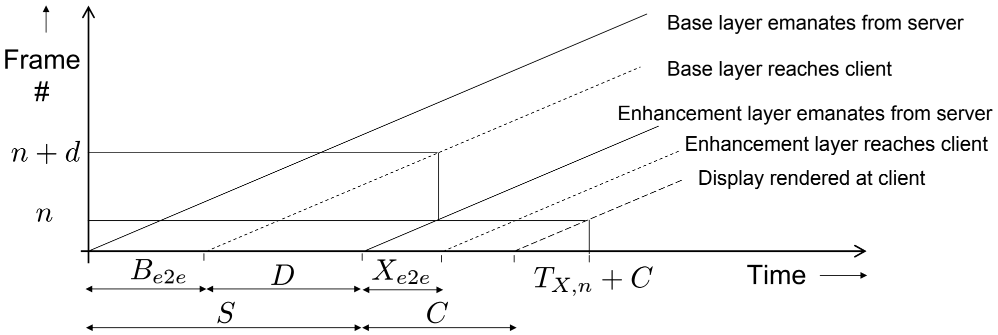
Frame #
$n+d$
n
Base layer emanates from server
Base layer reaches client
Enhancement layer emanates from server
Enhancement layer reaches client
Display rendered at client
B_{e2e}
D
X_{e2e}
$T_{X,n}+C$
Time
S
C

Fig. 2. Time-line for video streaming with interactive region-of-interest. The parameters of the design ensure that the ROI can be rendered instantly upon user input. The diagram is drawn for $S = D + B_{e2e}$.

tential parents. To monitor the online status of parents, peers send Hello messages regularly to their parents and the parents reply back. Since most tree disconnections are graceful and occur due to ROI change, the interval for sending Hello messages can be large to limit the protocol overhead. Similar to [6, 7], we also incorporate a loop-avoidance mechanism for every distribution tree.

4. EXPERIMENTAL RESULTS

We use the *Cardgame* sequence[1] having 3584x512 pixels and 25 frames/sec. Fig. 1 shows a frame of the sequence. It is a 360° panoramic video sequence stitched from several camera views. The camera setup is stationary and only the four card players move. We encode the thumbnail version at resolution 896x128 with an intraframe period of 15 frames using two consecutive B frames between anchor frames. The ROI display is 480x240 pixels. The two resolution layers in the coded representation have 1752x256 pixels (matches zoom factor of 1) and 3584x512 pixels resolution. These are encoded using slice sizes 64x256 and 128x128 respectively; i.e., the first layer has 28 slices horizontally and 1 slice vertically, whereas the second layer has 28 slices horizontally and 4 slices vertically. Every frame of the base layer is coded as one slice, hence, the server hosts 141 trees corresponding to 141 slices. The user's zoom factor is restricted between 1 and 6. The PSNR @ bitrate for the thumbnail is about 39.1 dB @ 162 kbps. The total data rate for the thumbnail and the ROI required by each peer is about 900 kbps on average. The performance of the video-content-aware ROI predictors in [10] is very close to that of perfect ROI prediction. In this paper, we focus on the evaluation of our distributed IROI P2P protocol and assume an oracle for perfect ROI prediction for pre-fetching. The values of S, C and D correspond to 50, 20 and 40 frames respectively.

We implemented the distributed IROI P2P protocol within the NS-2 network simulator [11]. We created a tree topology for the backbone network. Peers are placed on the randomly chosen edge nodes of the backbone network. The backbone links are sufficiently provisioned with high capacity. The propagation delay of each network link is set to 5 ms, thus resulting in propagation delays of about 50 ms between two peers. The uplink and downlink capacities of each peer are set to 2 Mbps. We recorded 100 ROI trajectories constituting navigation paths of as many peers. Each 1-minute-long trajectory starts at a random location. The 1-minute-long video sequence is obtained by looping a set of 298 frames. Peers are on for the entire session. Since no cross-traffic is simulated and the churn is only due to ROI change, the simulations are repeatable and a duration of 1 minute suffices for our experiments.

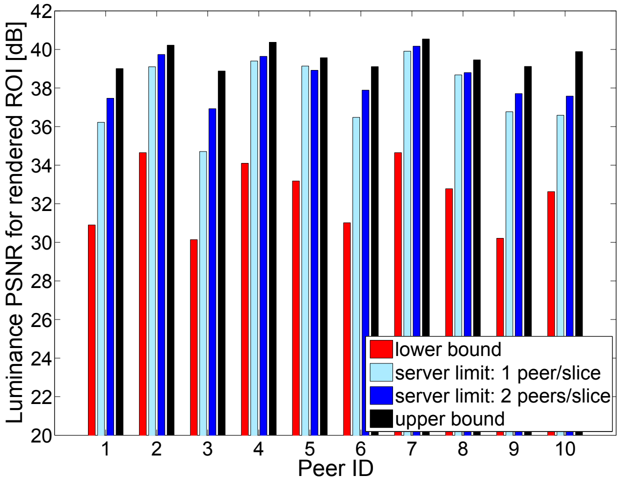

Fig. 3. Luminance PSNR for the rendered ROI shown for the first 10 peers. Peer population of 100 peers.

We limit the number of peers that the server can directly serve to either 1 or 2 for each slice. Without this restriction, the server's capacity might be exhausted and the system might not be able to supply a new slice that no peer currently subscribes. For the first 10 peers, Fig. 3 shows the average PSNR over an 800-frame-long interval starting from frame number 400. Also shown are the upper bounds and the lower bounds for the PSNR. The lower bound is the PSNR that results when no enhancement layer slices reach the peer, whereas the upper bound corresponds to the case when all required enhancement layer slices reach the client in time. The reference for the PSNR calculation is the ROI rendered from the original uncompressed multi-resolution video. With the limit of 2 peers for each slice, the average drop with respect to the upper bound is 1 dB, 0.7 dB and 0.2 dB for peer populations of 100, 30 and 15 peers respectively. With the limit of 1 peer for each slice, the average drop is 1.9 dB, 1 dB and 0.5 dB respectively. We observed no losses for the base layer. This is the result of the advancement of the base layer transmission which provides more time for the base layer packets to reach before their display deadlines. Moreover, we allow retransmissions for the base layer.

When the ROI changes, the peer might need to subscribe new slices. Fig. 4 shows the profile of number of new slices to subscribe

[1] Sequence provided by the Stanford Center for Innovations and Learning (SCIL), Stanford University, Stanford, CA.

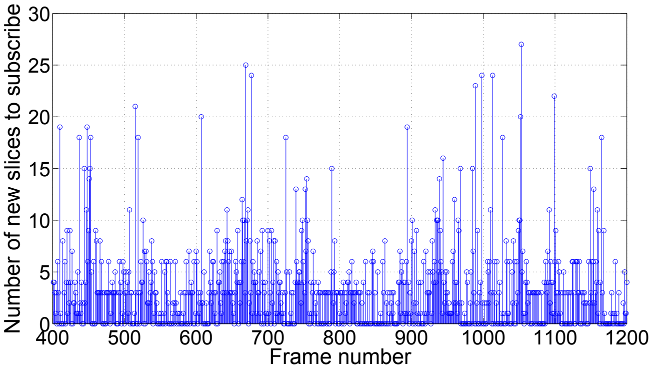

Fig. 4. Profile of number of new slices to subscribe due to ROI change, shown collectively for all 100 peers; does not include resubscriptions due to failing parents.

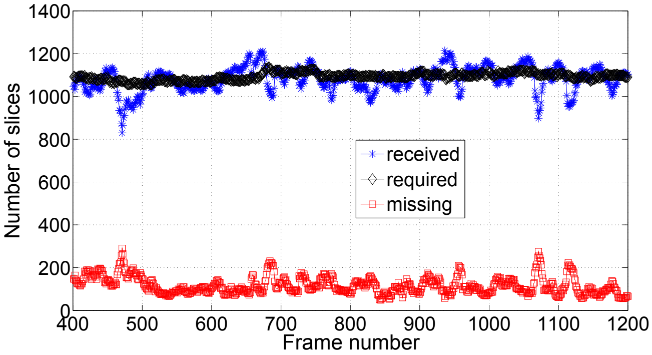

Fig. 5. Total number of required, received and missing slices, shown collectively for all 100 peers. The server is limited to directly serve 1 peer for each slice. Note that since we delay unsubscription slightly, it can result in more received slices than required.

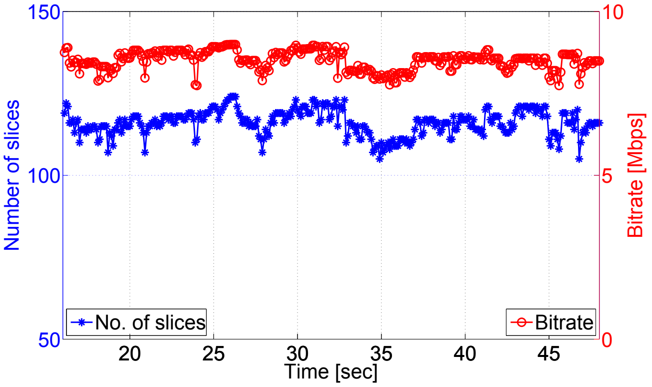

Fig. 6. Total load on the server with a population of 100 peers. The server is limited to directly serve 1 peer for each slice.

collectively for all 100 peers. Fig. 5 shows the profile of total number of required, received and missing slices for all 100 peers when we limit the server to directly serve 1 peer for each slice.

Figure 6 shows the load on the server for a population of 100 peers when it directly serves at most one peer for each slice. The average server load is around 8.5 Mbps. The server load is upper bounded by the rate of the multi-resolution ensemble, which is around 10 Mbps. Compared to a traditional client-server unicast architecture, we observe a 10.5x bit-rate reduction at the server on average. With peer populations of 75, 30 and 15, the reduction is around 7.7x, 3.7x and 2.2x respectively. With 15 peers, the average server load is about 7 Mbps. This indicates that, in a certain regime, increasing the number of peers increases the load on the server and the bandwidth savings grow slowly in this regime. However, once the server load is close to the upper bound, the bandwidth savings grow faster with increasing number of peers.

Our distributed IROI P2P protocol results in control traffic which constitutes less than 5% of the total traffic. About 65% of the control traffic results from probing potential parents.

5. CONCLUSIONS

We have designed a P2P multicast IROI video streaming system that allows a population of peers to interactively watch regions of a high-spatial-resolution video. Every peer enjoys the control of virtual pan/tilt/zoom yet our P2P overlay exploits the overlap among the ROIs for efficient and scalable delivery. The topology of our P2P overlay adapts to the changing ROIs. With 100 peers we obtain more than 10x bandwidth reduction with respect to client-server unicast IROI for a compromise of less than 2 dB in video quality. Once the number of peers is large enough to saturate the load on the server, the bandwidth savings grow faster with the number of peers. We employ pre-fetching of the ROI to meet the stringent latency constraint. Further improvements in our system will reduce the gap in quality with respect to traditional client-server unicast systems while retaining the bandwidth savings of the P2P approach.

6. REFERENCES

[1] “SopCast,” online: http://www.sopcast.org/.

[2] “PPLive,” online: http://www.pplive.com/.

[3] “Coolstreaming,” online: http://www.coolstreaming.us/.

[4] “GridMedia,” online: http://www.gridmedia.com.cn/.

[5] E. Kurutepe, M. R. Civanlar, and A. M. Tekalp, “Interactive transport of multi-view videos for 3DTV applications,” *Journal of Zhejiang University , SCIENCE A 7(5)*, 2006.

[6] E. Setton, J. Noh, and B. Girod, “Rate-distortion optimized video peer-to-peer multicast streaming,” *Proc. of Workshop on Advances in Peer-to-Peer Multimedia Streaming at ACM Multimedia*, pp. 39–48, Nov. 2005, invited paper.

[7] E. Setton, P. Baccichet, and B. Girod, “Peer-to-peer live multicast: A video perspective,” *Proceedings of the IEEE*, vol. 96, no. 1, pp. 25–38, Jan. 2008.

[8] A. Mavlankar, P. Baccichet, D. Varodayan, and B. Girod, “Optimal slice size for streaming regions of high resolution video with virtual pan/tilt/zoom functionality,” *Proc. of 15th European Signal Processing Conference (EUSIPCO), Poznan, Poland*, Sept. 2007.

[9] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, “Splitstream: High-bandwidth content distribution in a cooperative environment,” *Proc. of IPTPS, Berkeley, CA*, Feb. 2003.

[10] A. Mavlankar, D. Varodayan, and B. Girod, “Region-of-interest prediction for interactively streaming regions of high resolution video,” *Proc. of 16th Intl. Packet Video Workshop, Lausanne, Switzerland*, pp. 68–77, Nov. 2007.

[11] “The Network Simulator - ns-2,” online: www.isi.edu/nsnam/ns.