WPI-CS-TR-01-02 March 2001

Improving Multimedia Streaming with Content-Aware Video Scaling

by

Avanish Tripathi
Mark Claypool

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Abstract

Streaming video applications on the Internet generally have very high bandwidth requirements and yet are often unresponsive to network congestion. In order to avoid congestion collapse and improve video quality, these applications need to respond to congestion in the network by deploying mechanisms to reduce their bandwidth requirements under conditions of heavy load. In reducing bandwidth, video with high motion will look better if all the frames are kept but the frames have low quality, while video with low motion will look better if some frames are dropped but the remaining frames have high quality. Unfortunately current video applications scale to fit the available bandwidth without regard to the video content. In this paper, we present a content-aware scaling mechanism that reduces the bandwidth occupied by an application by either dropping frames (temporal scaling) or by reducing the quality of the frames transmitted (quality scaling). We have designed a streaming video client and server with the server capable of quantifying the amount of motion in an MPEG stream and scaling each scene either temporally or by quality as appropriate, maximizing the appearance of each video stream. We have evaluated our setup by conducting a user study wherein the subjects rated the quality of video clips that were first scaled temporally and then by quality in order to establish the optimal mechanism for scaling a particular stream. We find that our content-aware scaling can improve video quality by as much as 50%.

1 Introduction

The Internet disseminates enormous amounts of information for a wide variety of applications all over the world. As the number of active users on the Internet has increased so has the tremendous volume of data that is being exchanged between them, resulting in periods of transient congestion on the network. To overcome short-term congestion and avoid long term congestion collapse, various congestion control strategies have been built into the Transmission Control Protocol (TCP), the de facto transport protocol on the Internet. For multimedia traffic however, TCP is not the protocol of choice. Unlike traditional data flows, multimedia flows do not necessarily require a completely reliable transport protocol because they can absorb a

limited amount of loss without significant reduction in perceptual quality [4]. Also, multimedia flows have fairly strict delay and delay jitter requirements. Multimedia flows generally use the User Datagram Protocol (UDP). This is significant since UDP does not have a congestion control mechanism built in, therefore most multimedia flows are unable to respond to network congestion and adversely effect the performance of the network as a whole. By some estimates [2], about 77% of the data bytes accessed on the Web are in the form of multimedia objects.

While proposed multimedia protocols like TFRC [5] and RAP [16] respond to congestion by scaling back the data rate, they still require a mechanism at the application layer to map the scaling technique to the data rate. In times of network congestion, the random dropping of frames by the router[6] [11] may seriously degrade multimedia quality since the encoding mechanisms for multimedia generally bring in numerous dependencies between frames [14]. For instance, in MPEG encoding, dropping an independently encoded frame will result in the following dependent frames being rendered useless since they cannot be displayed and would be better off being dropped also rather that occupying unnecessary bandwidth. A multimedia application that is aware of these data dependencies can drop the frames that are the least important much more efficiently than can the router [7]. Such application specific data rate reduction is called *media scaling*.

A fine grained content-based packet forwarding mechanism [17] has been developed for differentiated service networks. This mechanism assigns relative priorities to packets based on the characteristics of the macroblocks contained within it. These characteristics include the macroblock encoding type, the associated motion vectors, the total size in bytes and the existence of any picture level headers. Their proposed scheme requires RED/RIO queue management and weighted fair queuing to provide the differentiated forwarding of packets with high priorities and therefore will not work in today's Internet.

A simple mechanism that uses temporal scaling for MPEG streams is suggested in [3]. In case of congestion, the frame rate is reduced by dropping frames in a predefined precedence (first B-frames and then P-frames) until the lowest frame rate, where only the I-frames are played out, is reached or the minimum bandwidth requirement matches the availability. An adaptive MPEG Streaming player based on similar techniques was developed at the the Oregon Graduate Institute of Science and Technology [19]. These systems

have the capabilities for dynamic rate adaptation but do not support real-time, automatic content detection. Automatic adaptive content-based scaling may significantly improve the perceptual quality of their played out streams.

The above mechanisms, while considering the specific characteristics of streaming flows, do not take into account the content of the video flows when scaling bandwidth. Media scaling techniques for video can be broadly categorized as follows [1]:

- *Spatial scaling:* In spatial scaling, the size of the frames is reduced by transmitting fewer pixels and increasing the pixel size, thereby reducing the level of detail in the frame.

- *Temporal scaling:* In temporal scaling, the application drops frames. The order in which the frames are dropped depends upon the relative importance of the different frame types. In the case of MPEG, the encoding of the I-frames is done independently and they are therefore the most important and are dropped last. The encoding of the P-frames is dependent on the I-frames and the encoding of the B-frames is dependent on both the I-frames and the P-frames, and the B-frames are least important since no frames are encoded based upon the B-frames. Therefore, B-frames are most likely to be the first ones to be dropped.

- *Quality scaling:* In quality scaling, the quantization levels are changed, chrominance is dropped or DCT and DWT coefficients are dropped. The resulting frames are of a lower quality and may have fewer colors and details.

It has been shown that the content of the stream can be an important factor in influencing the choice of the preferred scaling technique (i.e. temporal, spatial or quality) [1]. For instance, if a movie scene has quick motion and had to be scaled then it would look better if all the frames were played out albeit with lower quality. That would imply the use of either quality or spatial scaling mechanisms. On the other hand, if a movie scene has little motion and needed to be scaled it would look better if a few frames were dropped but the frames that were shown were of high quality. Such a system has been suggested in [9] but the quantitative benefits to multimedia quality for the users has yet to be determined.

[20] has developed a filtering mechanism for multimedia applications capable of scaling media streams. Using these filters it is possible to change the characteristics of audio or video streams by dropping frames, dropping colors, changing the quantization levels etc. We utilize these filtering mechanisms in conjunction with a real-time content analyzer we developed that measures the motion in an MPEG stream in order to implement a content-aware scaling system. We conduct a user study where the subjects rate the quality of video clips that are first scaled temporally and then by quality in order to establish the optimal mechanism for scaling a particular stream. We find our scaling system can improve perceptual quality of video by as much as 50%.

The remainder of this paper is organized as follows. Section 2 talks about the related work in this field, Section 3 discusses the methodology and approach of our work including our motion measurement technique, Sections 4 and 5 detail our experiments and their results, respectively and Section 6 describes our conclusions and possible future work.

2 Related Work

Various mechanisms have been proposed for multimedia protocols to respond to congestion on the Internet.

TFRC [5] is a mechanism for equation-based congestion control for unicast traffic. Unlike TCP, TFRC refrains from reducing the sending rate in half in response to a single packet-loss. Therefore, traffic such as best-effort unicast streaming multimedia could find use for this TCP-friendly congestion control mechanism. A TCP-friendly protocol called MPEG-TFRCP [15] was implemented and evaluated for fairness in bandwidth distribution among the TCP and the MPEG-TFRCP flows. RAP [16] is a TCP-friendly Rate Adaptation Protocol, which employs an additive increase, multiplicative decrease (AIMD) algorithm. Its primary goal is to be fair and TCP-friendly while separating network congestion control from application level reliability. Our content aware video-scaling can make the most effective use of bandwidth from these protocols.

Another approach to media scaling uses a layered source coding algorithm [13] with a layered transmission system [12]. By selectively forwarding subsets of layers at constrained network links, each user may receive the best

quality signal that the network can deliver. In the RLM (Receiver-driven Layered Multicast) scheme suggested, multicast receivers can adapt to the static heterogeneity of link bandwidths and dynamic variations in network capacity. However, this approach may have problems with excessive use of bandwidth for the signaling that is needed for hosts to subscribe or unsubscribe from multicast groups and fairness issues in that a host might not receive the best quality possible on account of being in a multicast group with low-end users.

A semi-reliable protocol that uses a TCP congestion window to pace the delivery of data into the network has also been suggested to handle multimedia congestion [8]. However other TCP algorithms, like retransmissions of dropped packets, etc. that are detrimental to real time multimedia applications have not been incorporated.

The above are a few of the network-centric approaches to solving the problems of unresponsiveness in multimedia flows but they do not consider the application level constraints of multimedia flows like frame interdependence and stream content.

3 Approach

In order to successfully develop a system that makes scaling decisions based upon the amount of motion in the video stream, we develop an automated means of measuring the amount of motion in the stream in real-time and then integrate this with the filtering system. The whole system is then capable of making content-aware decisions for the choice of the scaling mechanism to use for a particular sequence of frames. In the next two subsections we describe the motion measurement module and the filtering module of the system.

3.1 Motion Measurement

In our system, we have used an MPEG video stream to explore our approach. The MPEG video compression algorithm relies on two basic techniques: block-based motion compensation for reduction of temporal redundancy and transform domain-(DCT) based compression for reduction of spatial redundancy [10]. Prediction and interpolation are used for motion compensation.

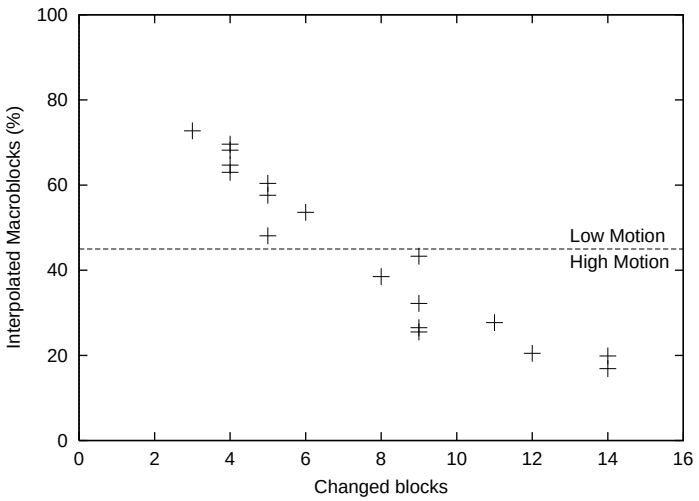

Figure 1: Motion Measurement

Motion-compensated prediction assumes that locally the current picture can be modeled as a translation of the picture at some previous time. In the temporal dimension, motion-compensated interpolation is a multi-resolution technique: a sub-signal with a low temporal resolution (typically 1/2 or 1/3 of the frame rate) is coded and the full-resolution signal is obtained by interpolation of the low-resolution signal and the addition of a correction term.

A typical MPEG stream contains three types of frames: Intra-encoded frames(I), Predicted frames (P) and Interpolated frames (B-for Bidirectional prediction). Each frame is further decomposed into 16x16 blocks called macroblocks, the basic motion-compensation unit. All macroblocks in the I-frames are encoded without prediction and the I-frame is thus independent of any other frames. The macroblocks in the P-frame are encoded with forward prediction from references made from previous I-frames and P-frames or may be intra-coded. Macroblocks in B-frames may be coded with forward prediction from past I-frames or P-frames, with backward prediction from future I-frames or P-frames, with interpolated prediction from past and future I-frames or P-frames or they may be intra-coded.

Our system uses the percentage of interpolated macroblocks in the B-frames as a measure of motion. A high number of interpolated macroblocks implies that a greater portion of the frame is similar to frames that are already

existing in the stream (i.e. less motion) and a low number of interpolated macroblocks implies that there are a greater number of changes between frames (i.e. more motion). To test the effectiveness of this measure of motion we conducted a pilot study. We encoded 18 video clips, each 10 seconds long and containing no scene changes. For each clip we divided the frames into 16 equal blocks and counted the number of blocks whose content changed during the clip. The percentage of interpolated macroblocks in the MPEG clip was then computed using *mpeg_stat* [18], an MPEG analysis tool.

Figure 3 shows the graph obtained when we plot the percentage of interpolated macroblocks against the number of blocks in which changes were observed when viewing the video clips. The x-axis shows the number of blocks that were observed to change during the movie clip and the y-axis shows the percentage of interpolated macroblocks for the corresponding clip. We notice that movies that had a higher number of blocks that changed (implying more motion) have a lower percentage of interpolated macroblocks and those with a lower number of changed blocks (implying less motion) have a high percentage of interpolated macroblocks. Although coarse, this methodology seems to work well when making decisions regarding scaling policies.

For our system, we need to categorize the sequence of frames into two categories, low motion or high motion. Sequences having greater than 45% interpolated macroblocks are classified as low motion and those having less than 45% are classified as high motion. This classification may be made more fine grained as the need arises.

Figure 3.1 shows the variation of the motion value for computations made every 1, 4 and 8 frames over an interval of 80 frames. This clip has an average interpolated macroblock value of 60% over its entire duration. While the variation is too high when the value is computed with every frame, there is not a significant increase in the smoothness of the curve for computations done every 8 frames compared to computations made every 4 frames. Therefore, in order to respond to changes in the amount of motion we compute the motion value for every 4 frames served. This parameter can also be varied to change the granularity of the system. Further evaluation of our measure of motion we leave as future work.

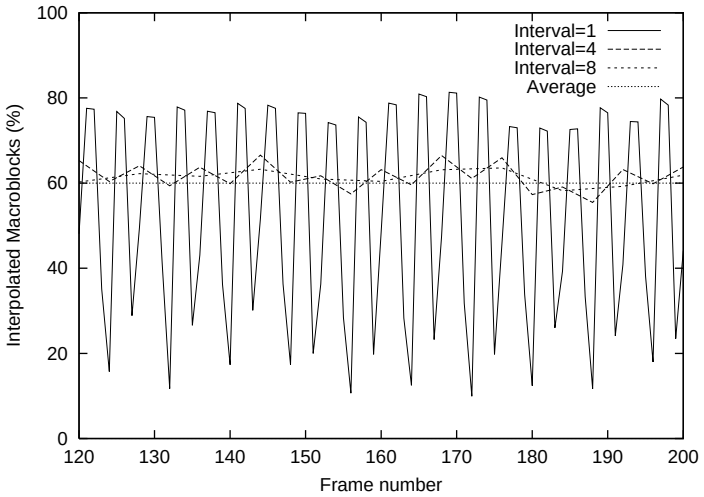

Figure 2: Motion Computation Interval

Table 1: Scale Levels

Scaling Level	Level	Scaling Method	Frame Rate (fps)	Bandwidth(%)
None	N/A	N/A	30	100
Temporal	1	No B frames	13	70
Temporal	2	No P or B frames	5	11
Quality	1	Requant Q = 7	30	65
Quality	2	Requant Q = 31	30	10

3.2 Filtering Mechanisms

[21] have developed a filtering system that operates on compressed video and can perform temporal and quality scaling. For temporal scaling we use the media discarding filter that has knowledge of frame types (eg. I, P or B) and can drop frames to reduce the frame rate thereby reducing the bandwidth. For quality scaling, we use the re-quantization filter. It operates on semi-compressed data, i.e. it first de-quantizes the DCT-coefficients and then re-quantizes them with a larger quantization step. As quantization is a lossy process the bit-rate reduction results in a lower quality image.

For our experiments we use five distinct scale levels. Table 1 shows the different scales and their corresponding frame-rate and bandwidth.The first

being full quality and frame rate and two levels each of temporal and quality scaling. Each temporal scaling method corresponds to a quality scaling method with a similar bit-rate reduction.

4 Experiments

We conducted a user study in order to verify the effectiveness of our content-aware scaling system. For the user study we had 22 undergraduate and graduate students from the Computer Science department in our school. We encoded 18 video clips from a cross-section of television programming. All the clips were approximately 10 seconds in duration and did not have scene changes. Using our measure of motion described in Section 3.1, we categorized these movies as having either high motion or low motion.

All the video clips were shown on 3 identical systems that had Pentium III processors, 128 MB of RAM running Linux. The clips were present on the local hard drives. They were shown with the following scaling levels (as shown in Table 1): full quality; no B-frames (temporal scaling, level 1); no B-frames or P-frames (temporal scaling, level 2); re-quantization factor set to 7 (quality scaling, level 1); and re-quantization factor set to 31 (quality scaling, level 2). We selected 2 clips from each category and asked the subjects to rate the 5 differently scaled versions of each of the clips. To rate the perceptual quality of the clips the subjects were asked to assign a number between 1 and 100 with 1 being the lowest quality and 100 being the highest quality.

For each clip, we calculated the mean rating with a 90% confidence interval.

5 Result Analysis

Figure 3 shows the graph we obtain when we plot the user perceived quality against the different scaling levels for a low motion clip. It shows four men talking at a bar while they have their drinks. This clip has an average of 70% interpolated macroblocks over the entire 10 second duration. We observe that temporal scaling does consistently better than quality scaling for the low motion clip. We also observe that with quality scaling the user perceived quality drops linearly but with temporal scaling the perceived quality drops

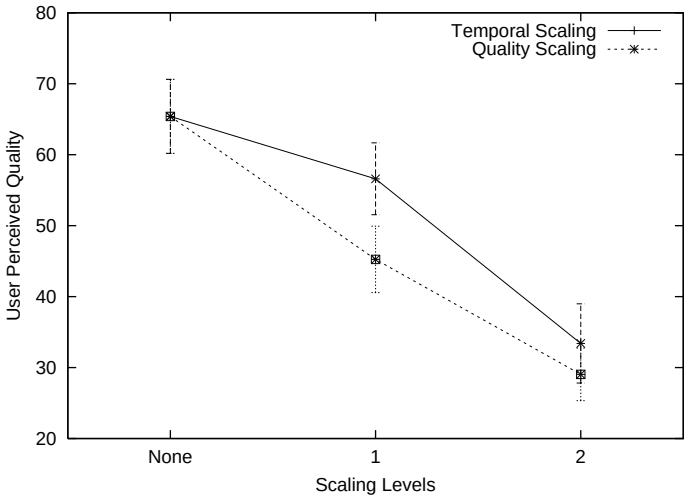

Figure 3: Low Motion Clip (70% Interpolated Macroblocks)

more rapidly as the frame rate reduces. We suspect there is a threshold below which users find the perceived quality unacceptable, and when the frame rate drops below this threshold smooth movement is lost. We expect this number to be between 4 to 8 frames per second, and we are currently working on more fine grained scaling levels to accurately determine this frame rate.

Figure 4 shows a similar graph for the clip having 57% interpolated macroblocks on an average over the whole clip. This is also a low motion clip having more than 45% interpolated macroblocks. It shows a character from the popular television sitcom "Friends" as she talks on the phone while walking across a room. Here again temporal scaling is consistently better than quality scaling and the user perceived quality drops sharply for the low frame rate of 5 frames per second.

Figure 5 shows the graph that we obtain for a high motion clip that shows a man riding a horse as he tries to catch a bull. It has 27% interpolated macroblocks on an average over the whole clip. As expected, we observe that quality scaling performs consistently better than temporal scaling. We also observe that the drop in user perceived quality for temporal scaling level 2 is not as pronounced as in previous graphs probably because the users found temporal scaling as a whole (and not just for low frame rates at level 2) to be inappropriate for high motion videos.

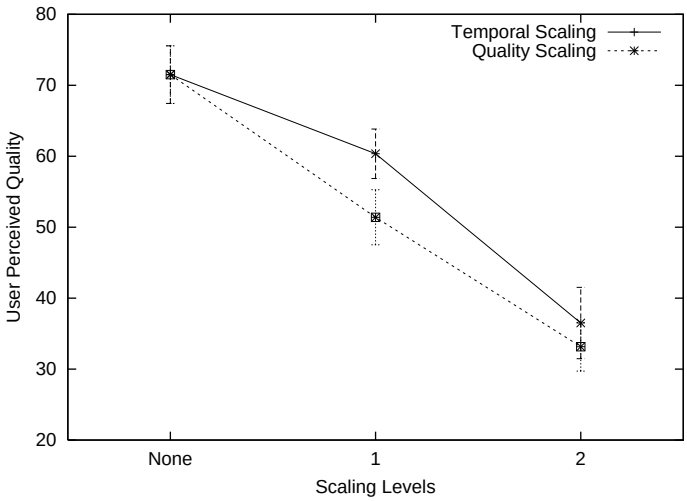

Figure 4: Low Motion Clip (57% Interpolated Macroblocks)

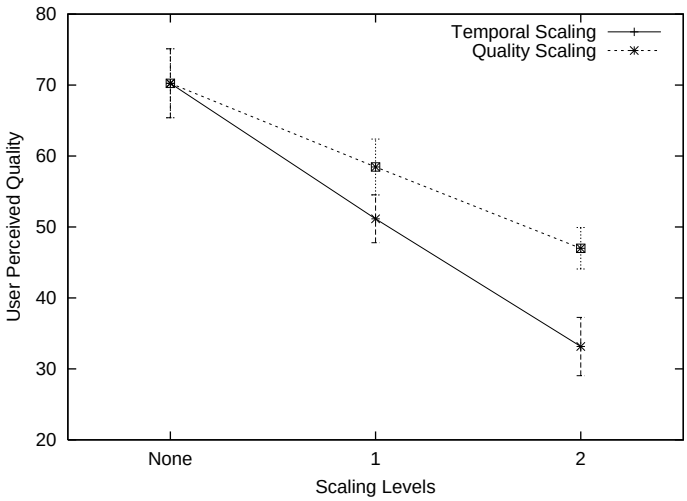

Figure 5: High Motion Clip (27% Interpolated Macroblocks)

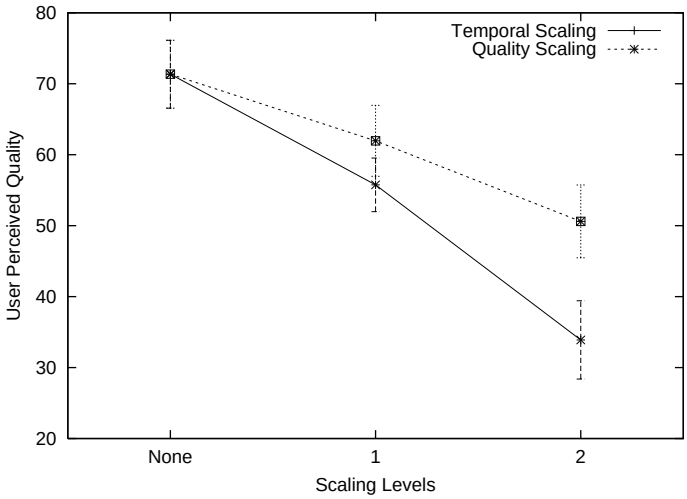

Figure 6: High Motion Clip (20% Interpolated Macroblocks)

We obtain a similar graph in Figure 6 for a high motion clip, a car commercial, having an average of 20% interpolated macroblocks. As before, quality scaling is consistently better to users than temporal scaling for this high motion clip.

6 Conclusions and Future Work

In this paper we have presented an application level solution to the problem of congestion due to unresponsive multimedia streams on the Internet. By introducing responsiveness at the application layer we eliminate the need for random dropping of packets due to congestion at the routers. This is significant in the case of multimedia streams because there are numerous dependencies between frames and losing packets from one frame might result in other frames being rendered useless. Our system takes into account the content of the video stream when choosing the scaling technique in order to have the minimum possible drop in quality for the end user.

We have implemented a system to quantify the amount of motion in a video stream and used it to design a content-aware scaling system for video. Our system determines the optimal scaling technique to apply when the available bandwidth does not permit us to serve the stream at full quality.

We verify our methodology by conducting a user study to determine the user perception of video quality after scaling the stream.

The results from the user-study show that if a movie clip with less motion is to be scaled to reduce the bandwidth it consumes it must be scaled temporally whereas a high motion movie clip must be scaled by quality to have the best user perceived quality after scaling. Our experiments have shown that the improvement in user perceived quality can be as much 50% when we scale using our content-aware technique.

We are currently working on a system that adaptively scales video streams in real time, taking into consideration the available bandwidth and the video content of the stream. The system will have more fine grained scaling levels than the ones used in the experiments described in this paper. This will also help us to accurately determine the threshold (in frames per second) below which temporal scaling leads to unacceptable user perceived quality. We will also measure the benefits of our system on video with fluctuating amounts of motion.

References

[1] P. Bocheck, A. Campbell, S.-F. Chang, and R. Lio. Utility-based Network Adaptation for MPEG-4 Systems. In *Proceedings of Ninth International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, June 1999.

[2] S. Chandra and C. Ellis. JPEG Compression Metric as a Quality Aware Image Transcoding. In *Proceedings of Second Usenix Symposium on Internet Technologies and Systems (USITS '99)*, October 1999.

[3] J. Chung and M. Claypool. Better-Behaved, Better-Performing Multimedia Networking. In *Proceedings of SCS Euromedia Conference (COMTEC)*, May 2000.

[4] M. Claypool and J. Tanner. The Effects of Jitter on the Perceptual Quality of Video. In *Proceedings of ACM Multimedia Conference*, volume 2, October 30 - November 5 1999.

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-Based Congestion Control for Unicast Applications. In *Proceedings of ACM SIGCOMM 2000*, August-September 2000.

[6] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, Aug. 1993.

[7] M. Hemy, U. Hangartner, P. Steenkiste, and T. Gross. MPEG System Streams in Best-Effort Networks. In *Proceedings of Packet Video Workshop'99*, April 1999.

[8] S. Jacobs and A. Eleftheriadis. Streaming Video using Dynamic Rate Shaping and TCP Congestion Control. *Journal of Visual Communication and Image Representation, Special Issue on Image Technology for World Wide Web Applications*, 9(3):211–222, September 1998.

[9] C. Kuhmunch, G. Kuhne, C. Schremmer, and T. Haenselmann. Video-Scaling Algorithm Based on Human Perception for Spatio-temporal Stimuli. In *Proceedings of SPIE Multimedia Computing and Networking (MMCN)*, volume 4312, January 2001.

[10] D. LeGall. MPEG- A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):47–58, April 1991.

[11] D. Lin and R. Morris. Dynamics of Random Early Detection. In *Proceedings of ACM SIGCOMM Conference*, Sept. 1997.

[12] S. McCanne, V. Jacobsen, and M. Vetterli. Receiver-driven Layered Multicast. In *Proceedings of ACM SIGCOMM Conference*, Aug. 1996.

[13] S. McCanne, M. Vetterli, and V. Jacobson. Low-complexity Video Coding for Receiver-driven Layered Multicast. *IEEE Journal on Selected Areas in Communications*, 16(6):983–1001, August 1997.

[14] J. Mitchell and W. Pennebaker. *MPEG Video: Compression Standard.* Chapman and Hall, 1996. ISBN 0412087715.

[15] M. Miyabayashi, N. Wakamiya, M. Murata, and H. Miyahara. Implementation of Video Transfer with TCP-Friendly Rate Control Protocol. In *Proceedings of International Technical Conference on Cir-*

cuits/Systems, Computers and Communications (ITC-CSCC 2000), July 2000.

[16] R. Rejaie, M. Handley, and D. Estrin. RAP: An End-to-End Rate-Based Congestion Control Mechanism for Realtime Streams in the Internet. In *Proceedings of IEEE Infocom '99*, March 1999.

[17] J. Shin, J. Kim, and C. J. Kuo. Content-Based Video Forwarding Mechanism in Differentiated Service Networks. In *Proceedings of International Packet Video Workshop*, May 2000.

[18] University of California, Berkeley. Berkeley MPEG-1 Video Analyzer : mpeg-stat. Interent site `http://bmrc.berkeley.edu/frame/research/mpeg/`.

[19] J. Walpole, R. Koster, S. Cen, C. Cowan, D. Maier, D. McNamee, C. Pu, D. Steere, and L. Yu. A Player for Adaptive MPEG Video Streaming over the Internet. In *Proceedings of 26th Applied Imagery Pattern Recognition Workshop, SPIE*, October 1997.

[20] N. Yeadon, F. Garcia, and D. Hutchinson. Filters: QoS Support Mechanisms for Multipeer Communications. *IEEE Journal on Selected Areas in Communications*, 14(7):1245–1262, September 1996.

[21] N. Yeadon, F. Garcia, D. Hutchinson, and D. Shepherd. Continuous Media Filters for Heterogeneous Internetworking. In *Proceedings of SPIE Multimedia Computing and Networking (MMCN'96)*, January 1996.