1-7-2004

Adaptive Video Streaming using Content-Aware Media Scaling

Mark Claypool
Worcester Polytechnic Institute, claypool@wpi.edu

Avanish Tripathi
Worcester Polytechnic Institute

WPI-CS-TR-04-01 January 2004

Adaptive Video Streaming using Content-Aware Media Scaling

by

Mark Claypool and Avanish Tripathi

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

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

Adaptive Video Streaming using Content-Aware Media Scaling

Mark Claypool and Avanish Tripathi
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609, USA
(508) 831-5357
{claypool}@cs.wpi.edu

January 7, 2004

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 quality 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 scaled by quality in order to establish the optimal mechanism for scaling a particular stream. We find that our content-aware scaling can improve perceived video quality by as much as 50%. We have also evaluated the practical impact of adaptively scaling the video stream by conducting a user study for longer video clips with varying amounts of motion and available bandwidth. We find that for such clips the improvement in perceptual quality on account of adaptive content-aware scaling is as high as 30%

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 [6]. Also, multimedia flows have fairly strict delay and delay jitter requirements. For these reasons, streaming video applications often use the User Datagram Protocol (UDP) rather than TCP. This is significant since UDP does not have a congestion control mechanism built in, meaning most multimedia flows are unable to respond to network congestion and adversely affect the performance of the network as a whole.

While proposed multimedia protocols [9, 19, 4] respond to congestion by scaling back the data rate, these protocols 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 congested routers may seriously degrade multimedia quality since the encoding mechanisms for multimedia generally bring in numerous dependencies between frames [16]. For instance, in MPEG encoding, dropping an independently encoded frame (I-frame) will result in the following dependent frames (P-frames or B-frames) not being fully decoded, so it may be more effective to just discard the frames before sending them rather that occupying unnecessary bandwidth. In fact, a 3% packet loss in an MPEG coded bit stream can translate into a 30% frame error rate [3]. A multimedia application that is aware of these data dependencies can discard the frames that are the least important much more efficiently than can the router [11]. Such application specific data rate reduction is called *media scaling.*

Media scaling techniques for video can be broadly categorized as follows [2]:

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

- *Temporal scaling:* In temporal scaling, the sending discards frames. The order in which the frames are discarded 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 discarded last. The encoding of the P-frames is dependent on the I-frames, 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 discarded.

- *Quality scaling:* In quality scaling, the quantization levels are reduced, chrominance is dropped or compression coefficients are dropped. The resulting frames are lower in 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) [2]. For instance, if a movie scene has high 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 had 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 [13] but the quantitative benefits to multimedia quality for the users has yet to be determined. Other techniques for multimedia scaling have been proposed (see Section 2), which operate at the network layer or the application layer or at both the layers. Unfortunately, none of the techniques take into account the content of the video when scaling bandwidth.

In this work, we utilized filtering mechanisms [23] to change the characteristics of audio or video streams by discarding frames and changing the quantization levels 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 conducted 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 that content-aware scaling can improve perceptual quality of video by as much as 50%. We evaluated the performance of the adaptive scaling system by conducting a user study where the users watched video clips that had varying amounts of motion as opposed to the relatively consistent amounts of motion for the earlier user study. We find that adaptive content-aware scaling can improve the perceptual quality of video by as much as 30%.

The remainder of this paper is organized as follows: Section 2 describes related work in this field; Section 3 discusses our methodology and approach, including our motion measurement technique; Sections 4 and 5 detail our experiments and their results, respectively; and Sections 6 and 7 describe our conclusions and possible future work.

2 Related Work

Various techniques have been proposed to address the problem of network congestion from unresponsive multimedia streams on the Internet. These techniques can be broadly classified as being network level, application level or a hybrid of both. In this section, we describe some of the proposed techniques from all three classes.

2.1 Network Level Techniques

TCP-Friendly Rate Control (TFRC) [9] is a mechanism for equation-based congestion control for unicast traffic over the Internet. Unlike TCP, where the sending rate is controlled by a congestion window that is halved for every lost packet, TFRC refrains from reducing the sending rate in half in response to a single packet-loss. Instead, the sender explicitly adjusts its sending rate as a function of measured rate of loss events, where a loss event consists of one or more packets lost in a single round trip time. TFRC has been shown to provide smoother data rates than does TCP while still providing TCP-Friendly data rates in the presence of congestion.

The Rate Adaptation Protocol (RAP) [19] is a TCP-friendly protocol that employs an additive increase, multiplicative decrease (AIMD) algorithm for congestion control. RAP also includes an architecture for the delivery of real-time layered encoded stored real-time streams over the Internet [18]. RAP's primary goal is to be TCP-friendly while separating network congestion control from application level reliability and error control because the

former depends on the state of the network while the latter is application specific. Thus, unlike TCP, RAP does not offer a 100% reliable transport layer which, within bounds, is acceptable to multimedia applications.

Above are a few of the network-centric approaches to addressing the network bandwidth problems of unresponsiveness in multimedia flows, but they do not consider the application level constraints of multimedia flows like frame interdependence and stream content. Our approach can use the above protocols and allow better use of the available network bandwidth.

2.2 Application Level Techniques

Shin et al [20] proposes a content-based packet video forwarding mechanism for a differentiated services (DiffServ) [1] network. The QoS interaction between the video applications and the DiffServ network is taken into account. The interaction is performed through a dynamic mapping between the relative priority score (RPS) of each video packet and the differentiated packet forwarding mechanism. The RPS is computed taking into account the characteristics of its component macroblocks like the encoding type, associated motion vectors, total size in bytes, etc. The RPS of each packet is then mapped to one of the network DiffServ levels. Each packet is then assigned to a queue class which gets a specific reliability level depending, possibly, on the price paid for the service. The differentiation in queuing can potentially be realized by adopting multiple queues with different drop curves known as multiple RED [7] or RED with in and out bit (RIO) [5]. However, multiple queue management with packet priorities is not achievable in todays Internet and even a progression towards such priority classes will entail a single best-effort class of traffic for the foreseeable future.

Yeadon et al [24] develop a filtering mechanism for multimedia applications that is capable of scaling media streams, predominantly MPEG-1 and Motion-JPEG encoded streams. Most of these filters work on compressed or semi-compressed bit-streams and can change the characteristics of the multimedia streams by dropping frames, dropping colors, changing quantization levels, etc. We integrate these filters with our server module and use them in conjunction with a real-time content analyzer we developed to build our adaptive content-aware scaling system.

Walpole et al [22] develop a player for adaptive MPEG video streaming over the Internet. The player is capable of adapting to the available bandwidth by scaling the stream temporally (i.e. discarding frames at the sender in a predefined precedence), taking advantage of the inherent characteristics of the MPEG encoding scheme as shown in Table 1. The first frames to be discarded in case of congestion are the bi-directional encoded (B-frames) since the other (I- and P-frames) do not depend on the B-frames for their decoding. The predictive encoded (P-frames) are discarded next. [4] uses a similar temporal scaling scheme to develop a flow controlled multimedia application over UDP.

2.3 Hybrid Techniques

Jacobs and Eleftheriadis [12] propose a semi-reliable protocol that uses a TCP congestion window to pace the delivery of data into the network to manage multimedia congestion. However other TCP algorithms, like retransmissions of dropped packets, etc. that are

Table 1: Temporal Rate Adaptation for MPEG

Frame Rate	Send Pattern												
2.5	I	-	-	-	-	-	-	-	-	-	-	-	I
5.0	I	-	-	P	-	-	-	-	-	-	-	-	I
10.0	I	-	-	P	-	-	P	-	-	P	-	-	I
15.0	I	-	-	P	B	-	P	-	-	P	B	-	I
20.0	I	-	B	P	-	B	P	-	B	P	-	B	I
30.0	I	B	B	P	B	B	P	B	B	P	B	B	I

detrimental to real time multimedia applications have not been incorporated.

Receiver-driven Layered Multicast (RLM) [14] uses a layered source coding algorithm [15] with a layered transmission system. In RLM, the source signal is encoded into a number of layers that can be incrementally combined to provide progressive refinement of the received signal. The layers of the signal are multicast on distinct channels. The RLM receivers subscribe to different layers by subscribing to different multicast channels. 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.

MPEG-TFRCP [17] is another TCP-friendly protocol that has been developed to support video traffic over the Internet. This protocol achieves fairness among TCP and UDP connections by adjusting the sending rate to the estimated TCP throughput at regular intervals of duration 32 times the round trip time between the sender and the receiver.

The network condition (expressed by the *round-trip time* (RTT) and the packet loss probability) is estimated from the feedback information obtained by means of ACK packets. The video sending rate is then adjusted against the target rate by choosing an appropriate quantizer scale (i.e. using quality scaling).

The network level approaches do not consider the application level constraints of multimedia flows like frame interdependence and stream content. The application level techniques for media scaling do take into consideration the specific characteristics of the multimedia streams but none are content-aware. It has been shown that video content plays an important part in determining the optimal scaling mechanism for a video stream [2]. For instance, in the case of high-motion scenes, spatial scaling or quality scaling techniques are more suitable than temporal-domain scaling techniques (i.e. dropping of frames) because the details within a frame may not be as important in high-motion conditions. In contrast, low-motion scenes favor the opposite approach. Since there is little change between successive frames in a low-motion scene, dropping frames does does not degrade perceptual quality if the remaining frames are shown at full resolution. Such a system has been suggested in [13] but the quantitative benefits to multimedia quality for the users has yet to be determined.

Since the recipient of an improved video stream via the Internet is a human observer, we evaluate the benefits of our adaptive content-aware scaling system by conducting user studies.

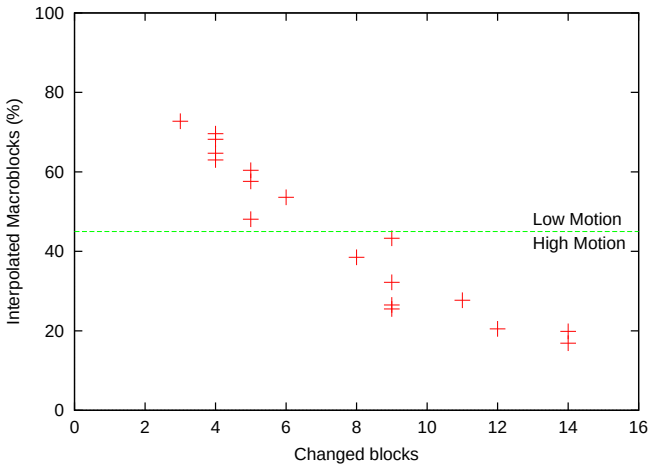

Figure 1: Motion Measurement

3 Approach

In order to successfully develop an adaptive content-aware scaling system, we developed an automated means of measuring the amount of motion in the stream in real-time and then integrated this with the filtering system. The whole system was then capable of making content-aware decisions in choosing the scaling mechanism to use for a particular sequence of frames. In the next three Subsections we describe the motion measurement module, the filtering module and describe the functionality of the full system, respectively.

3.1 Motion Measurement

Although our techniques are applicable to most video compression schemes, 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. Motion-compensated prediction assumes that the current picture frame can be modeled as a translation of a picture frame 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- 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

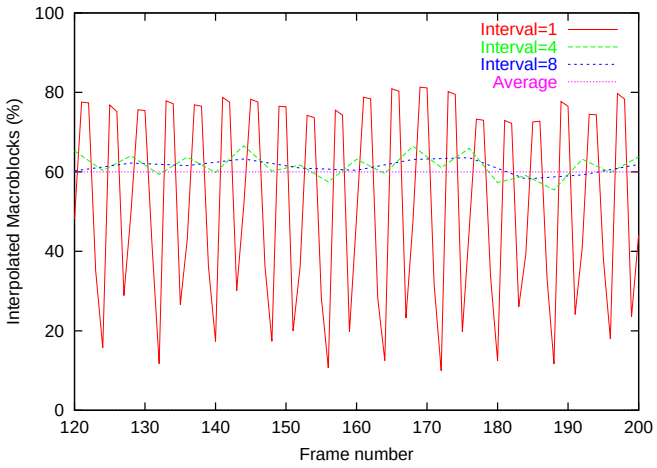

Figure 2: Motion Computation Interval

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 of a variety of content, 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 visually changed during the clip. The percentage of interpolated macroblocks in the MPEG clip was then computed using *mpeg_stat* [21], an MPEG analysis tool. Figure 1 depicts the percentage of interpolated macroblocks versus 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 measure of motion provides quantitative information on the amount of visual motion in the current frame sequence for 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 2 shows the variation of the motion values in a clip 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

Table 2: Scale Levels for User Study 1

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

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 and with alternate frame sequences. Further evaluation of our measure of motion we leave as future work.

3.2 Filtering Mechanisms

[24] have developed a filtering system that operates on compressed video and can perform temporal and quality scaling. We extend their filtering system and integrate it with our content-aware scaling system. For temporal scaling we use the media discarding filter that has knowledge of the frame type (i.e. I-, P- or B-frame) and can discard frames to reduce the frame rate, thereby reducing the bandwidth. For quality scaling, we use the re-quantization filter that 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 first set of experiments (user study 1) we have defined three distinct scale levels. For the second set of experiments (user study 2) we increase the number of scale levels to four. Table 2 shows the different scales and their corresponding frame-rates and bandwidth for the experiments for the first user study. Since we compare temporal scaling and quality scaling in our first user study it is important that the scale levels have similar post-filter bandwidths. The first row in Table 2 shows the clips at encoded quality and frame rate (30 frames per second). We then have two levels (and corresponding rows) each of temporal and quality scaling. Each temporal scaling method corresponds to a quality scaling method with a similar bit-rate reduction.

3.3 Adaptive Content-Aware Media Scaling System

In addition to evaluating the benefits of content-aware scaling on the perceptual quality of video streams that have consistent motion characteristics, we designed and implemented the adaptive content-aware scaling system. Figure 3 shows the architecture of our system.

The system consists of 4 distinct modules: *server*, *filter*, *network* and the *client*.

- *Server:* The server in the system takes as input an MPEG file, parses and packetizes it and streams it over the network to the client. The server is also capable of quantifying the amount of motion in the video stream by using the motion measurement sub-module.

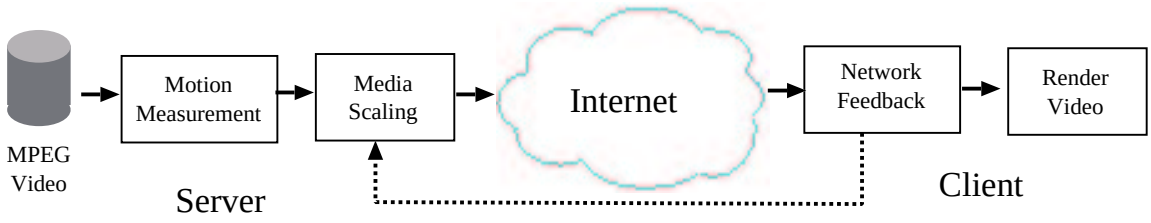

Figure 3: Adaptive Content-Aware Media Scaling System Architecture

- *Filter Module:* Upon obtaining a measure of motion, the filter module scales the video, as appropriate. The filter module has two kinds of filters: *Temporal Filter* and *Quality Filter*. The temporal filter is a frame discarding filter and scales the video in the temporal domain. The quality filter is a re-quantization filter and scales the video stream in the quality domain.

- *Network Module:* The network module resides on the client side and monitors the congestion in the network by recording packet loss rate and round trip time. Dropped packets (i.e. from packet loss) are detected by a gap in the packet sequence numbers. Round-trip time is recorded by reflecting a time-stamp back to the server. In the event of congestion (some packet loss), the feedback module will send control messages (with the current packet loss rate and latest time stamp) to the server. The server computes a TCP-Friendly bandwidth rate based on [8].

- *Client:* The client module is a regular MPEG decoder that is capable of playing out frames that are received over network sockets.

3.3.1 System Functionality

Figure 4 shows the sequence of steps that take place in the system. When the server is activated it blocks, waiting for control messages at a predefined port number. The filter module also blocks for control messages at a different port number upon activation (Step 1). When the user at the client side wishes to play a video, the client sends a request to the server with the name of the MPEG file (Step 2). Upon receiving the request the server begins reading the file off the disk, packetizes it and passes it on to the filter module (Step 3). In the absence of congestion the filter module simply forwards these packets over the network on a UDP connection to the client (Step 13).

In case of network loss the network module at the client sends a control message to the server indicating a reduction in available bandwidth. The server then invokes the motion measurement module to obtain the amount of motion in the video in the scene being served at that particular instant of time (Step 5). Depending upon the amount of motion, the server invokes the appropriate filter to reduce the bandwidth occupied by the stream (i.e. quality filter for a high motion scenes and the temporal filter for a low motion scene) (Steps 6 through 11).

The system uses 4 distinct scaling levels (used in user study 2) as shown in Table 3.

```
(1) Activate Server and Filter
(2) Receive Movie Request from Client
(3) while not (end_of_file(movie_file)) {
(4)   Parse and send to Filter Module
(5)   if (congestion) Measure Motion
(6)     if (highmotion)
(7)       Invoke Quality Filter
(8)         send Quality scaled
(9)     else
(10)      Invoke Temporal Filter
(11)        send Temporally scaled
(12)    else
(13)      send full quality frames
(14) }end of while
```

Figure 4: Server Algorithm

4 Experiments

We conducted two user studies in order to evaluate the effectiveness of our adaptive media scaling system. In the first user study we evaluate the potential benefits of content-aware scaling and in the second user study we evaluate the potential benefits from our adaptive content-aware scaling system for streams with variation in their motion characteristics and for different network bandwidth fluctuation rates.

Both user studies were conducted on identical systems with Pentium III 600 MHz processors and 128 MB of memory running Linux 2.2.14. The video clips were present on the local hard drives of each of the systems so that actual network conditions did not influence the video quality and instead, induced network load could be controlled by our system. Users rated the clips on a scale of 1 to 100 with 100 being the highest quality.

For the first user study, we encoded 18 MPEG video clips from a cross-section of television programming. All the clips were approximately 10 seconds in duration and consisted of a single scene in order to have consistent motion characteristics. Using our measure of motion, we categorized these clips as having either high motion or low motion. We selected two clips from each category, and each of the four video clips was shown with the following five scaling types and levels (as shown in Table 2): full quality; no B-frames (temporal scaling, level 1); no B-frames or P-frames (temporal scaling, level 2);

Table 3: Scale Levels for User Study 2

Scaling Type	Level	Scaling Method	Frame Rate (fps)	Bandwidth(%)
None	N/A	N/A	30	100
Temporal	1	Alternate B frames dropped	21	85
Temporal	2	All B frames dropped	13	70
Temporal	3	No P or B frames	5	11
Quality	1	Requant Q = 4	30	85
Quality	2	Requant Q = 7	30	65
Quality	3	Requant Q = 31	30	10

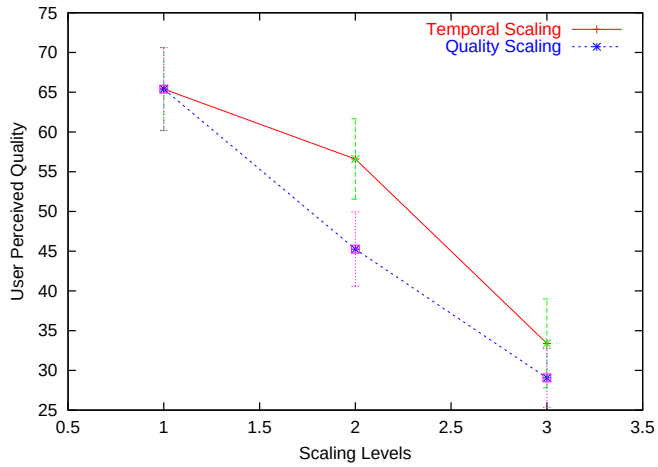

Figure 5: Low Motion Clip (70% Interpolated Macroblocks)

re-quantization factor set to 7 (quality scaling, level 1); and re-quantization factor set to 31 (quality scaling, level 2).

For the second user study, we encoded 2 clips with varied motion characteristics. Each of the clips was approximately 25 seconds in duration and had one scene change where a transition from low motion to high motion or vice versa took place. Depending upon the amount of motion in the currently being displayed scene and the available bandwidth, the system automatically selected the most appropriate scaling technique.

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

5 Analysis

In this section we present the results of our evaluations of the benefits of content-aware scaling and our adaptive content-aware scaling system.

5.1 Content-Aware Scaling

For the first set of experiments (user study 1) we used four 10 second clips: two having high motion and two having low motion.

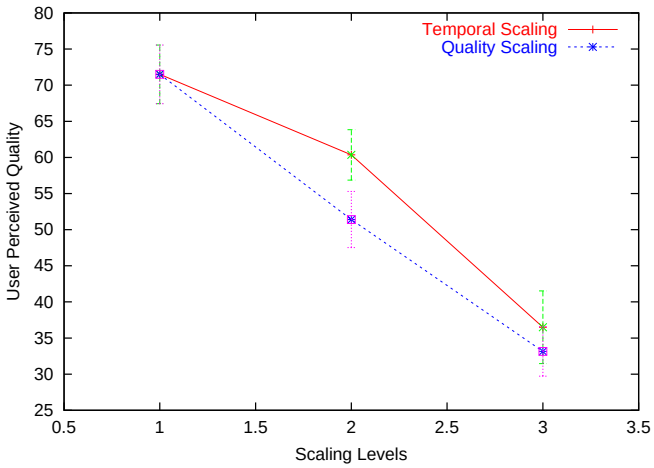

Figure 6: Low Motion Clip (57% Interpolated Macroblocks)

Figure 5 shows the graph we obtain when we plot the user perceived quality against the different scaling levels for a low motion clip[1]. This clip has an average of 70% interpolated macroblocks over the entire 10 second duration. From Figure 5, temporal scaling provides consistently better than quality scaling for the low motion clip. With quality scaling the user perceived quality drops linearly but with temporal scaling the perceived quality drops 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 6 shows a similar graph for the clip having 57% interpolated macroblocks on an average over the whole clip[2]. From Figure 6, 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 7 shows the graph that we obtain for a high motion clip[3] with 27% interpolated macroblocks on an average over the whole clip. For this clip, quality scaling performs consistently better than temporal scaling. 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 8 shows the graph that we obtain for another high motion clip[4]. with an average of 20% interpolated macroblocks. As in the previous high motion clip, quality scaling is consistently better to users than temporal scaling for this high motion clip.

[1]The clip is of four men talking at a bar while having a drink.

[2]The clip is of a character from the popular television sitcom "Friends" as she talks on the phone while walking across a room.

[3]The clip is of a man riding a horse as he tries to catch a bull.

[4]The clip is a car commercial with a driving scene.

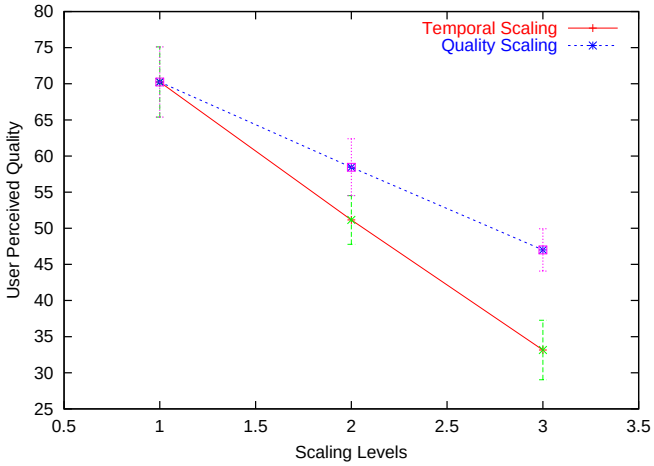

Figure 7: High Motion Clip (27% Interpolated Macroblocks)

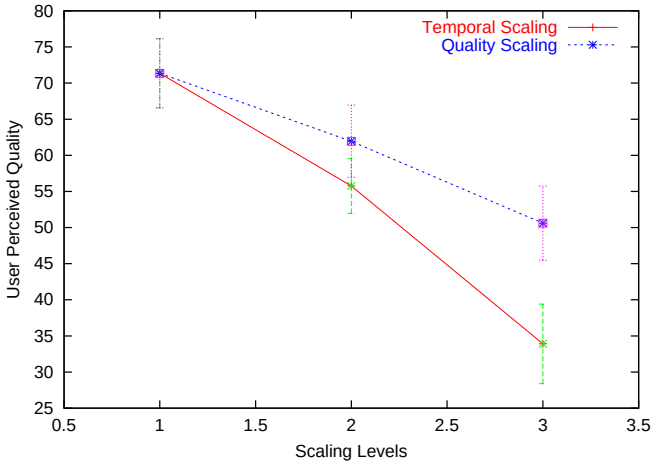

Figure 8: High Motion Clip (20% Interpolated Macroblocks)

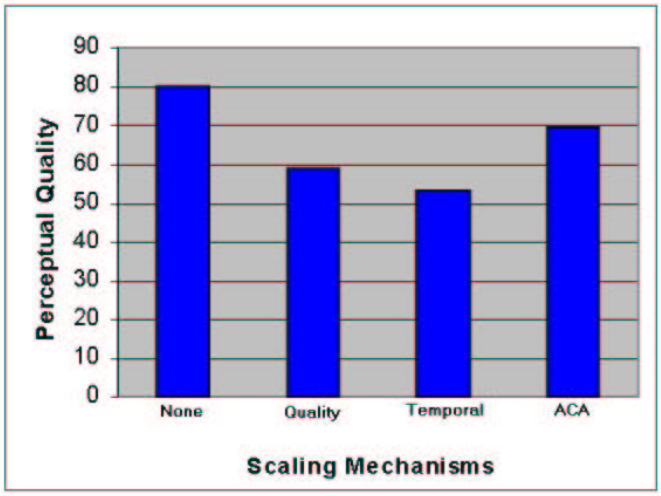

Figure 9: Clip 5- Bandwidth changes every 2s

5.2 Adaptive Content-Aware Scaling

In this section, we present the results of our second set of experiments (user study 2). For this study we used two video clips. The clips were approximately 25 seconds in duration and had one scene change each where the transition between high motion to low motion or vice versa takes place. Clip 5 shows a scene from a talk show (low motion) followed by a car commercial (high motion). Clip 6 shows a scene from the television sitcom,*Friends* (predominantly low motion), followed by a commercial for an adventure show (predominantly high motion). Clip 6 has considerably more variation in the motion values than clip 5.

Figures 9 through 12 show the graphs we obtain when we plot the perceived quality of clips 5 and 6 against different scaling mechanisms for varying bandwidths. In all the graphs, perceived quality is plotted on the y-axis and scaling mechanisms are plotted on the x-axis. On the x-axis, the column labeled *None* shows the average perceptual quality value for the clip at full quality without any scaling. The column labeled *Quality* shows the average perceptual quality when the clip is quality scaled. The column labeled *Temporal* shows the average perceptual quality when the clip is temporally scaled, and the column labeled ACA shows the perceptual quality when the clip is adaptively content-aware scaled.

Figure 9 shows the graph obtained when the available bandwidth changes every 2 seconds for clip 5. The 90% confidence interval for *None* is [78.4%-81.6%], for *Quality* is [55.8%-62.5%], for *Temporal* is [49.5%-56.4%] and for ACA is [66.1%-72.6%]. Figure 10 shows the graph when the bandwidth changes every 500ms for the same clip. For this graph, the 90% confidence interval for *None* is [78.4%-81.6%], for *Quality* is [51.6%-57.6%], for *Temporal* is [49.4%-55.6%] and for ACA is [69.1%-73.5%]. There is an appreciable improvement in the perceptual quality of the clip when using adaptive content-aware scaling compared to the case where the stream is scaled without regard to the content of the stream. The improvement is nearly 30% both when bandwidth changes every 2s and when the bandwidth changes every 500ms.

As shown in Figure 11, for clip 6, we find that there is an appreciable improvement in

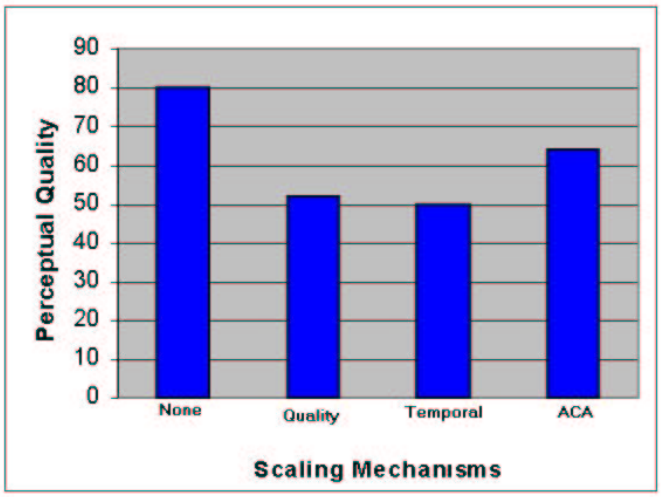

Figure 10: Clip 5- Bandwidth changes every 500ms

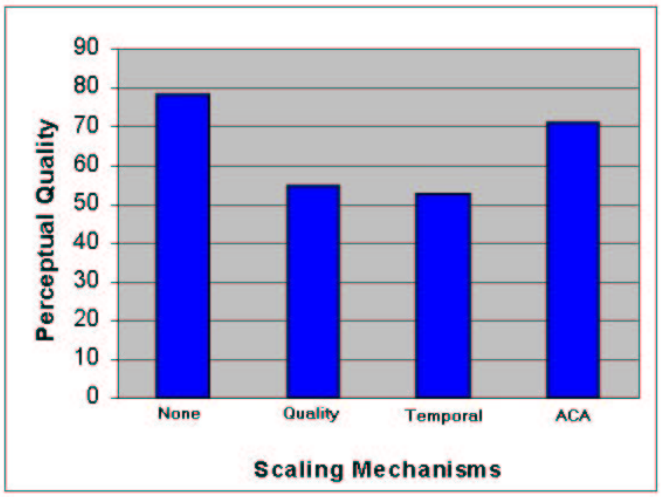

Figure 11: Clip 6- Bandwidth changes every 2s

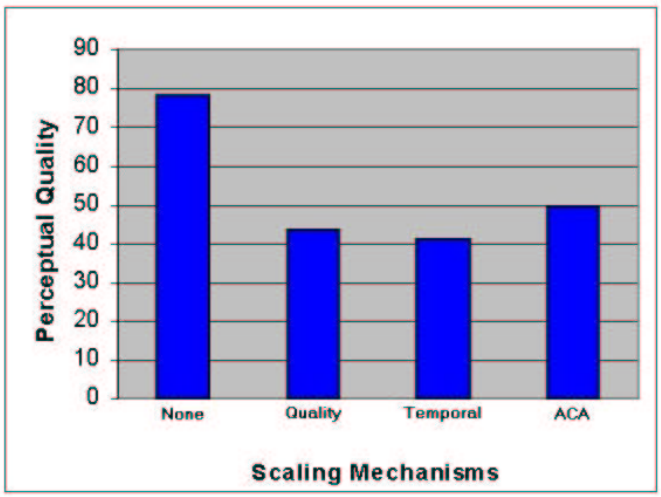

Figure 12: Clip 6- Bandwidth changes every 500ms

the perceptual quality when the available bandwidth changes every 2s. The 90% confidence interval for *None* is [71.6%-75.4%], for *Quality* is [48.8%-55.8%], for *Temporal* is [46.7%-53.8%] and for ACA is [61.9%-66.8%]. From Figure 12, the improvement is not as high when the bandwidth changes every 500ms. In this case the 90% confidence interval for *None* is [71.6%-75.4%], for *Quality* is [41.5%-45.7%], for *Temporal* is [38.3%-43.4%] and for ACA is [47.6%-51.3%]. This reduction in the improvement is probably because the frequent changes in motion characteristics of this clip cause the scaling type to also change very frequently (as often as 500ms). The frequent changes in the scaling type may be what causes the users to rate this clip lower for the 500ms case.

6 Conclusions

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 reduce 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 the key frames results in degradation in quality for other frames.

We have built an adaptive system that takes into account the content of the video stream when choosing the scaling technique in order to have the minimum possible drop in perceptual quality for the end user. The system performs the scaling operations in real-time as the video stream is served to the client.

We have shown that in order to maximize perceptual quality under constrained (TCP-Friendly) bandwidth, the amount of motion in a video stream must be considered when choosing a scaling mechanism for a video stream. For instance, a movie scene had a lot of motion and required scaling 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 had little motion and required scaling

it would look better if a few frames were discarded, but the frames that were shown were of high quality.

We have implemented a method to quantify the amount of motion in a video stream and used it to design the adaptive content-aware scaling system for video streams. Using the motion measurement system, our scaling system determines the optimal scaling technique to apply when the available bandwidth does not permit serving the stream at full quality. We verify our methodology by conducting two user studies to determine perceptual quality of the video stream after the stream has been scaled. Our experiments shown that the improvement in user perceived quality can be as much 50% when we scale using the content-aware technique for clips that have consistent motion characteristics over the entire duration of the clip.

We also conducted experiments to stream video clips with variations in motion characteristics and bandwidth. We find that when bandwidth changes occur on the order of a few seconds, the improvement in perceptual quality with adaptive content-aware scaling is as high as 30%. We also find that if the motion characteristics of the clip change rapidly and the bandwidth also changes on the order of hundreds of milliseconds, the improvement in perceptual quality is somewhat reduced by the high frequency of the changes in scaling type. The increase in perceptual quality in such cases is only about 5-10%.

7 Future Work

In our work we simulate the variations in available network bandwidth by using the bandwidth distribution function. By developing a more accurate function to model network bandwidth we may get a better insight into the performance on this system on the Internet. Eventually we would like to use this system to stream video over the Internet, suggesting possible user studies under various Internet conditions.

In the course of our experiments we noticed that below a certain frame rate (4-8 frames per second) temporal scaling leads to unacceptable perceptual quality. By accurately determining this threshold we can put a lower bound below which temporal scaling is ineffective. In such cases, quality scaling should be used instead of temporal scaling.

For our experiments, at any one point of time, we only use one scaling method (either quality or temporal). There may be a larger benefit to perceptual quality with hybrid scaling (i.e. combining temporal scaling with quality scaling). This could be specially useful when the amount of motion does not strictly fall into either the *high* or *low* categories. In addition, spatial scaling as well may have the most benefits for some movies under certain network conditions.

Finally, we could try some of the scaling methods used in our work for video streams on audio streams and evaluate their effectiveness for audio streams.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. *IETF Request for Comments (RFC) 2475*, December 1998.

[2] Paul Bocheck, Andrew Campbell, Shih-Fu Chang, and Raymond Lio. Utility-based Network Adaptation for MPEG-4 Systems. In *Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, June 1999.

[3] J. Boyce and R. Gaglianello. Packet Loss Effects on MPEG Video sent over the Public Internet. In *Proceedings of ACM Multimedia*, pages 181–190, Bristol, U.K., September 1998.

[4] Jae Chung and Mark Claypool. Better-Behaved, Better-Performing Multimedia Networking. In *Proceedings of SCS Euromedia*, May 2000.

[5] D. Clark and W. Fang. Explicit Allocation of Best-Effort Service. *IEEE/ACM Transactions on Networking*, August 1998.

[6] Mark Claypool and Jonathan Tanner. The Effects of Jitter on the Perceptual Quality of Video. In *Proceedings of the ACM Multimedia Conference*, volume 2, November 1999.

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

[8] Sally Floyd and Kevin Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, February 1999.

[9] Sally Floyd, Mark Handley, Jitendra Padhye, and Jorg Widmer. Equation-Based Congestion Control for Unicast Applications. In *Proceedings of ACM SIGCOMM Conference*, pages 45 – 58, 2000.

[10] Didier Le Gall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):46 – 58, 1991.

[11] Michael Hemy, Urs Hangartner, Peter Steenkiste, and Thomas Gross. MPEG System Streams in Best-Effort Networks. In *Proceedings of Packet Video Workshop*, April 1999.

[12] 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 WWW Applications*, 9(3):211–222, September 1998.

[13] Christoph Kuhmunch, Gerald Kuhne, Claudia Schremmer, and Thomas 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.

[14] Steven McCanne, Van Jacobsen, and Martin Vetterli. Receiver-driven Layered Multicast. In *Proceedings of ACM SIGCOMM Conference*, August 1996.

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

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

[17] Masaki Miyabayashi, Naoki Wakamiya, Masayuti Murata, and Hideo Miyahara. Implementation of Video Transfer with TCP-Friendly Rate Control Protocol. In *Proceedings of Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2000)*, July 2000.

[18] Reza Rejaie, Mark Handley, and Deborah Estrin. Architectural Considerations for Playback of Quality Adaptive Video over the Internet. Technical Report 98-681, CS Department, University of Southern California, November 1998.

[19] Rezza Rejaie, Mark 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*, 1999.

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

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

[22] Jonathan Walpole, Rainer Koster, Shanwei Cen, Crispin Cowan, David Maier, Dylan McNamee, Calton Pu, David Steere, and Liujin Yu. A Player for Adaptive MPEG Video Streaming Over The Internet. In *Proceedings of the SPIE Applied Imagery Pattern Recognition Workshop*, October 1997.

[23] 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.

[24] Nicholas Yeadon, Francisco Garcia, David Hutchinson, and Doug Shepherd. Continuous Media Filters for Heterogeneous Internetworking. In *Proceedings of SPIE Multimedia Computing and Networking (MMCN'96)*, January 1996.