Worcester Polytechnic Institute
Digital WPI

Masters Theses (All Theses, All Years) | Electronic Theses and Dissertations

2001-05-01

Adaptive Content-Aware Scaling for Improved Video Streaming

Avanish Tripathi
Worcester Polytechnic Institute

Follow this and additional works at: https://digitalcommons.wpi.edu/etd-theses

Repository Citation

Tripathi, Avanish, "*Adaptive Content-Aware Scaling for Improved Video Streaming*" (2001). *Masters Theses (All Theses, All Years)*. 605.
https://digitalcommons.wpi.edu/etd-theses/605

This thesis is brought to you for free and open access by Digital WPI. It has been accepted for inclusion in Masters Theses (All Theses, All Years) by an authorized administrator of Digital WPI. For more information, please contact wpi-etd@wpi.edu.

Adaptive Content-Aware Scaling for Improved Video Streaming

by

Avanish Tripathi

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

May 2001

APPROVED:

Professor Mark Claypool, Thesis Advisor

Professor Bob Kinicki, Thesis Reader

Professor Micha Hofri, Head of Department

If we knew what we were doing, it wouldn't be called research, would it?
Albert Einstein (1879-1955)

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 thesis, we present an adaptive 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 the impact of content-aware scaling 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 content-aware scaling can improve 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 also the improvement in perceptual quality on account of adaptive content-aware scaling is as high as 30%

Acknowledgments

As I write this last page in my report I cannot but look back at the past nine months that I have worked (or at least should have worked) on this thesis and I see a host of people without whom I could not have possibly completed what I had set out to do. Mark, I cannot thank you enough for all that you've done for me, not just for this thesis but everything to make my time here at WPI something to cherish.

Prof. Kinicki, thank you for being my reader and for all your comments and suggestions that helped make my work better. Prof. Wills, I am grateful for your help during the final user study for my thesis work.

I'd like to thank all my fellow graduate students who have done their theses for the hours spent trying to figure out why we were doing what we were doing and how it was all worth it. And to all my fellow students in the department who sat thorough my crude video clips to rate their hard to find quality, thank you !

I dedicate this thesis, my first piece of work of any consequence (or so I'd like to think) to my parents.

Contents

List of Figures

List of Tables

Chapter 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. The window based flow control approach of TCP, although extremely effective for congestion control, generally results in large variations in the bandwidth available to an application. Multimedia applications, on the other hand, require fairly consistent bandwidth availability. Also, unlike traditional data flows, multimedia flows do not necessarily require a completely reliable transport protocol like TCP because they can absorb a limited amount of loss and still achieve acceptable quality [CT99]. Further, multimedia flows have fairly strict delay and delay jitter requirements that are often violated by TCP's window based approach to data transmission and retransmissions.

For the reasons mentioned above multimedia flows generally use the User Datagram Protocol (UDP). This is significant since UDP does not have a congestion control mechanism built in, and therefore most multimedia flows are unable to respond to network congestion and adversely affect the performance of the network as a whole. By some estimates [CE99], about 77% of the data bytes accessed on the Web are in the form of multimedia objects like images, audio and video. Of this about 33% is streaming media that can potentially benefit from our proposed scaling technique.

While proposed multimedia protocols like TFRC [FHPW00] and RAP [RHE99] 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 packets by the router [FJ93] [LM97] may seriously degrade multimedia quality since the encoding mechanisms for multimedia streams generally bring in numerous dependencies between packets of different frames [MP96]. 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. In fact, a 3% packet loss in an MPEG coded bit stream can translate into a 30% frame error rate [BG98]. 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 that has no application level knowledge about the stream [HHSG99]. Therefore, an application level approach can be more effective and efficient for the bandwidth reduction of a multimedia flow. One such technique for application specific data rate reduction is called *media scaling*.

Media scaling techniques for video can be broadly categorized as follows [BCCL99]:

- *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 compression coefficients are dropped. The resulting frames are lower in quality and may have fewer colors and details.

Various application level multimedia scaling techniques have been proposed. A fine grained content-based packet forwarding mechanism [SKK00] 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 [CC00]. 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 [WKC+97]. These systems have the capabilities for dynamic rate adaptation but do not support real-time, automatic content detection. Adaptive content-aware scaling may significantly improve the perceptual quality of their played out streams.

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) [BCCL99]. For instance, if 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 dropped but the frames that were shown were of high quality. Such a system has been suggested in [KKSH01] but the quantitative benefits to multimedia quality for the users has yet to be determined.

[YGH96] has developed a filtering mechanism for multimedia applications capable of scaling media streams. Using these filters it is possible to change the characteristics of video streams by dropping frames, dropping colors, changing the quantization levels, etc. We utilize these filters in conjunction with a real-time content analyzer we developed that measures the motion in an MPEG stream, and design and implement an adaptive, content-aware video streaming system. We conducted a user study where 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 content aware scaling for clips that have consistently high motion or consistently low motion can improve the perceptual quality 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 the adaptive content-aware scaling system improves the perceptual quality of video by as much as 30 %.

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

Chapter 2

Related Work

Various techniques have been proposed to tackle the problem of network congestion from unresponsive multimedia streams on the Internet. These techniques can broadly be classified as being network level, application level or a hybrid of both. In this chapter we describe some of the proposed techniques from all the three classes (sections 2.1, 2.2, 2.3). In section 2.4 we summarize the related work and discuss the relevance of our work.

2.1 Network Level Techniques

The TCP congestion control mechanism in the Internet has been fairly successful in preventing congestion collapse. But the unsuitability of TCP for multimedia flows has also been widely recognized [FHPW00, RHE99, CC00]. This has prompted the development of numerous network level approaches based on the idea of building protocols that are TCP-friendly i.e. under similar network conditions they occupy no more bandwidth than a TCP flow would, but reduce factors that are detrimental to multimedia traffic. Therefore, traffic such as best-effort unicast streaming multimedia may benefit from a TCP-friendly protocol over TCP and perhaps even

UDP.

2.1.1 TCP-Friendly Rate Control

TCP-Friendly Rate Control (TFRC) [FHPW00] 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 uses an equation-based congestion control mechanism where a control equation gives the maximum acceptable sending rate as a function of the recent loss event rate. The sender adapts its sending rate, guided by this control equation, in response to the feedback from the receiver. The primary goal of this equation based congestion control mechanism is not to aggressively find and use available bandwidth, but to maintain a relatively steady sending rate while still being responsive to congestion.

2.1.2 Rate Adaptation Protocol

The Rate Adaptation Protocol [RHE99] is a TCP-friendly protocol that employs an additive increase, multiplicative decrease (AIMD) algorithm for congestion control. This paper presents an architecture for the delivery of real-time layered encoded stored real-time streams over the Internet [RHE98]. Its primary goal is to be fair and 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. Unlike TCP, RAP does not offer a 100% reliable transport layer which, within bounds, is acceptable to multimedia applications [CT99]. The server's transmission rate is continuously adjusted by the Rate Adaptation Protocol (RAP) in a TCP friendly fashion. The RAP module is exclusively in-charge of congestion control and loss detection. The layer manager adapts the quality of the transmitted streams based on the rate specified by the RAP module.

2.2 Application Level Techniques

In this section we briefly describe some of the application level techniques proposed for media scaling.

2.2.1 Content-based Packet Forwarding for Differentiated Service Networks

[SKK00] proposes a content-based packet video forwarding mechanism for a differentiated services (DiffServ) [BBC+98] 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. Under packetized video transmission, the relative priority assignment for a packet would be best if it can precisely represent its error propagation effect to the video quality at the receiver. For a video stream, a lost packet can lead to the content loss of subsequent packets due to temporal loss propagation as a result of the inter-block and inter-frame correlation. In fact, a raw packet loss rate of 3% in an MPEG encoded bit stream can result in a frame error rate as high as 30% [BG98]. The RPS is therefore 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 [FJ93] or RED with in and out bit (RIO) [CF98]. But the RED/RIO queue management and the weighted fair queuing scheduling will not work in todays Internet.

2.2.2 Filters for Multimedia Streams

[YGHS96] developed 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.

2.2.3 Temporal Scaling for MPEG

[WKC+97] developed 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. dropping frames at the sender in a predefined precedence. They take advantage of the inherent characteristics of the MPEG encoding scheme. The first frames to be dropped 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 dropped next. Rate adaptation is

Table 2.1: 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

accomplished by dropping a fraction of the frames of the different kinds as shown in the table 2.1. [CC00] uses a similar temporal scaling scheme to develop a flow controlled multimedia application over UDP.

2.3 Hybrid Techniques

In this section we describe some of the work that proposes systems at the network level as well as the application level.

2.3.1 Dynamic Rate Shaping

Jacobs and Eleftheriadis [JE98] have proposed 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 detrimental to real time multimedia applications have not been incorporated.

At the application level they use rate shaping for MPEG streams to match the bandwidth of the application to that of the network. This is done using signal processing techniques (i.e. quality scaling) that work on semi-compressed video to change the bandwidth of the stream.

2.3.2 Receiver-driven Layered Multicast (RLM)

RLM uses a layered source coding algorithm [MVJ97] with a layered transmission system [MJV96]. In this algorithm, the source signal is encoded into a number of layers that can be incrementally combined to provide progressive refinement of the received signal. By selectively forwarding subsets of layers at constrained network links i.e. forwarding only the number of layers that any link can manage, heterogeneity is managed by locally degrading the quality of the transmitted signal. The layers of the signal are multicast on distinct channels. The RLM receivers subscribe to different number of layers by subscribing to different multicast channels. The multicast receivers can therefore 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 un-subscribe 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.

2.3.3 MPEG-TFRCP

MPEG-TFRCP [MWMM00] 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 target rate of interval i (denoted by r_i), is determined as

$$r_i = \begin{cases} r_{TCP} \approx \frac{MTU}{RTT\sqrt{\frac{2p}{3}}+T_0 min\left(1,3\sqrt{\frac{2p}{3}}\right)p(1+32p^2)} & : \quad p > 0 \\ 2 \times r_{i-1} & : \quad p = 0 \end{cases}$$

where MTU stands for the maximum transfer unit size, p is the packet loss probability, RTT and T_0 are the round trip time and the retransmission timeout interval, respectively. The network condition (expressed by the 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 r_i by choosing an appropriate quantizer scale (i.e. using quality scaling).

2.4 Summary

In section 2.1 we described a few of the network level techniques to solving the problems of unresponsiveness in multimedia flows. But most of these approaches do not consider the application level constraints of multimedia flows like frame interdependence and stream content.

Most of the application level techniques and the hybrid techniques for media scaling described above 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 [BCCL99]. 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 [KKSH01] but the quantitative benefits to multimedia quality for the users is yet to be determined.

Keeping in mind that the data sink for a video transmission via Internet is a human observer, we evaluate the benefits of our adaptive content-aware scaling system by conducting user studies.

Chapter 3

Methodology

We followed the following methodology for developing our adaptive content-aware scaling system:

- Develop and verify the motion measurement mechanism for MPEG streams (Section 3.1.1)
- Define temporal and quality scaling levels such that corresponding levels occupy similar bandwidths (Section 3.1.2)
- Evaluate the potential impact of content-aware scaling on the perceptual quality of video by conducting a user study (Chapter 4)
- Build the complete scaling system that does adaptive content-aware scaling (Section 3.2)
- Evaluate the practical impact on perceptual quality of the complete system by conducting another user study (Chapter 4)

For ease of organization the evaluation of our system is described in a separate chapter (Chapter 4).

3.1 Content-Aware Scaling

In order to successfully develop a system that makes scaling decisions based upon the amount of motion in the video stream, we needed to develop an automated means of measuring the amount of motion in the stream in real-time and then integrate this with a filtering system. The whole system would then be 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 that we used to study the impact of content-aware scaling on the user-perceived quality of a video stream.

3.1.1 Motion Measurement

In the motion measurement system we developed, we have used an MPEG video stream to explore our approach. The next sub-section gives a brief overview of the MPEG compression standard.

MPEG Compression Standard

The Moving Picture Experts group (MPEG) developed the MPEG standard for the compression of moving video and associated audio for digital storage media at about 1.5 Mbps. The remainder of this section describes the video compression algorithms used in the standard.

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 [LeG91]. Motion compensation techniques are applied with both causal (pure predictive coding) and non-causal predictors (interpolated coding). The remaining signal (prediction

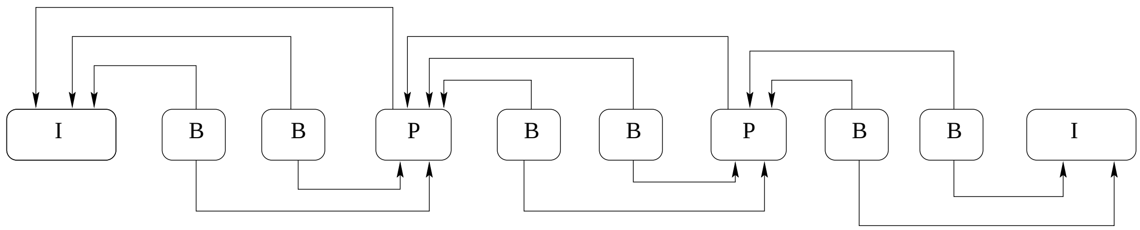

Figure 3.1: MPEG Frame Interdependency Relationships

error) is further compressed with spatial redundancy reduction (DCT). The information relative to motion is based on 16×16 blocks and is transmitted together with the spatial information. The motion information is compressed using variable length codes to achieve maximum efficiency.

Prediction and interpolation are used for motion compensation. 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). The frame inter-dependencies are illustrated in Figure 3.1. As the names suggest, the compression of the I-frame exploits only the spatial redundancy within the frame and is therefore independent of any other frame in the stream. I-frames provide access points for random access into the video stream but at the cost of lower compression rates. P-frames are coded with reference to a past frame (I or P) and will in general be used for future predicted pictures. B-frames provide the highest amount of compression but require both a past and a future reference for prediction. In addition, B-frames are never used as a reference.

In all cases when a picture is coded with respect to a reference, motion compensation is used to improve the coding efficiency. Each frame is further decomposed into 16×16 blocks called macroblocks. These macroblocks, as mentioned above, are the basic motion-compensation units. 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.

Motion Estimation

Our system uses the percentage of interpolated macroblocks in the B-frames over the duration of the stream 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 and the original signal may be reconstructed with pure interpolation. This in turn means that there are very few changes in the stream at that particular point of time. On the other hand, a low number of interpolated macroblocks implies that it is difficult to reconstruct the original signal with pure interpolation and either predictive or intra-encoding needs to be done. Therefore, a low number of interpolated macroblocks in the stream suggests that there is a greater number of changes between frames and hence more motion in the video stream.

To evaluate the effectiveness of this measure of motion we conducted a pilot study. We encoded 18 video clips selected from random television programming, where each video clip was approximately 10 seconds long. We also ensured that the video clips contained no scene changes since there is an increase in the number of interpolated macroblocks at the scene boundaries on account of changes in the content of the stream. As we viewed each clip, we visually divided the frames into 16 equal sub-blocks and counted the number of sub-blocks whose content changed over the duration of the clip. At the end of the clip this number was recorded. We then computed the percentage of interpolated macroblocks in the MPEG clip using *mpeg_stat* [Uni], an MPEG analysis tool. We then tried to correlate the amount of motion in the stream (from the changed sub-blocks counted earlier) and the percentage of interpolated macroblocks that we computed.

Figure 3.2 shows the graph obtained when we plot the percentage of interpolated macroblocks against the number of sub-blocks in which changes were observed when viewing the video clips. The x-axis in the graph is the number of sub-blocks that were observed to change during the movie clip and the y-axis is the percentage of interpolated macroblocks for the corresponding clip. It is fairly evident from the graph that movies that had a higher number of sub-blocks that changed (implying more motion) have a lower percentage of interpolated macroblocks and those with a lower number of changed sub-blocks (implying less motion) have a high percentage of interpolated macroblocks. This suggests that the percentage of interpolated macroblocks should be an effective measure to use when making decisions regarding scaling policies.

In order to prove that the amount of motion must influence the type of scaling to be used on a stream we start with a coarse categorization of the amount motion. For our system, we categorize the sequence of frames into either of the two categories, low motion or high motion. From visual inspection of Figure 3.2 we observe that the range of data points on the graph are almost evenly distributed above and below the 45% mark. Therefore, for our work sequences having greater than 45% interpolated

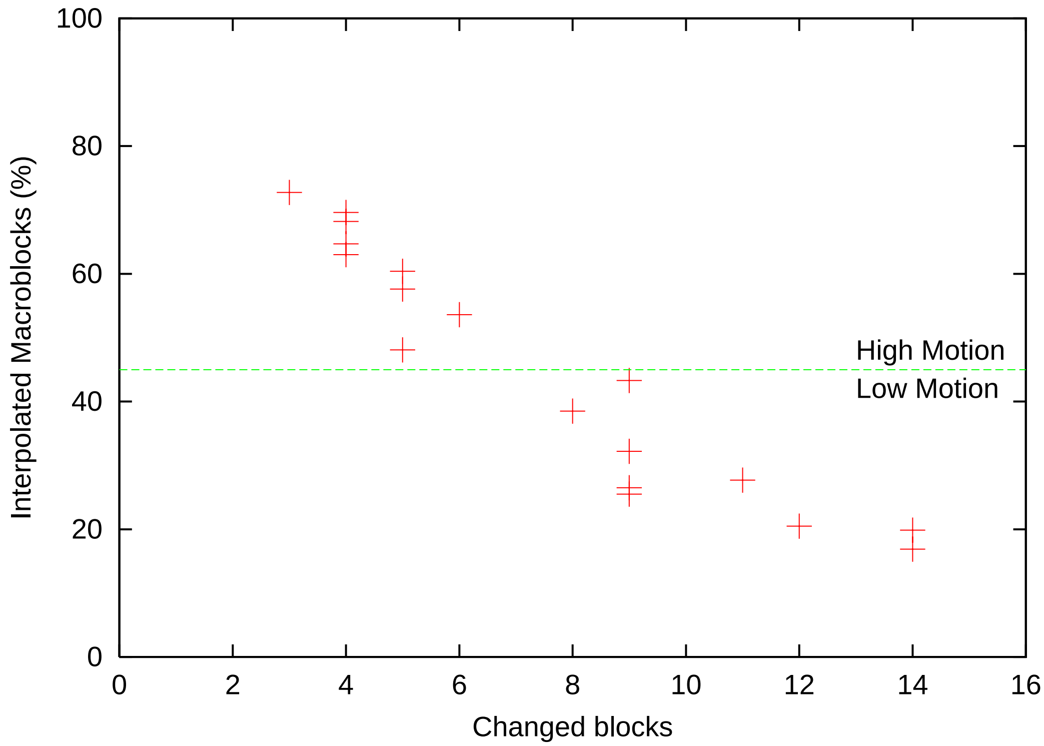

Figure 3.2: Motion Measurement

macroblocks are classified as low motion sequences and those having less than 45% are classified as high motion sequences. This classification may be made more fine grained for applications that require a greater number of motion levels.

In order to decide the frequency of the computations for the amount of motion we computed the values for intervals of 1 frame, 4 frames and 8 frames. Figure 3.3 shows the variation of the motion value for computations made every 1, 4 and 8 frames, respectively. The trade-off is the added computational load on the processor for computations done very frequently against the reduction in the responsiveness of the system or added delay for the client if computations are done for very large intervals. Moreover, since the number of interpolated macroblocks can be computed in real-time for every frame, our primary concern is on achieving the minimum number of frames to maximize responsiveness. In Figure 3.3, we show the computed

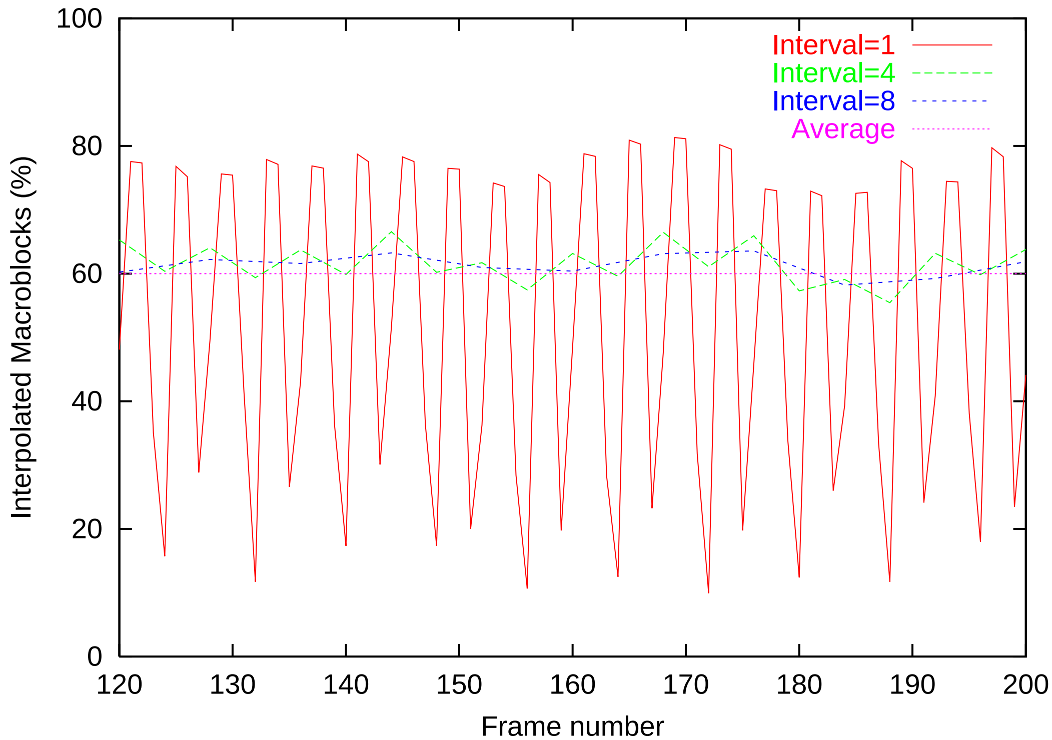

Figure 3.3: Motion Computation Interval

motion values for a window of 80 frames. This clip has an average interpolated macroblock value of 60% over its entire duration.

When the motion values are computed for every frame we find that the variation of the value is too great. The motion value crosses the assigned threshold of 45% almost every three frames. We therefore cannot make scaling decisions based on these values with any consistency. On the other hand when the computations are made every 8 frames we obtain a fairly smooth curve that rarely crosses our assigned threshold. Finally, when the computations are made every 4 frames the curve obtained is comparable to the curve for computations made every 8 frames. Therefore, in order to adequately respond to changes in the amount of motion we compute the motion value for every 4 frames served by the system to the client. This parameter can also be varied to change the granularity of the system. Further

Table 3.1: Scale Levels for User-Study 1

Scaling Type	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

evaluation of our measure of motion we leave as future work.

3.1.2 Filtering Mechanisms

As mentioned in Chapter 2, [YGHS96] developed a filtering system for multimedia streams. The filtering system operates on compressed and semi-compressed video and can be used to perform temporal and quality scaling. We extend the filtering system to integrate it with our adaptive content-aware scaling system. 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. By intelligently specifying the type of frames to drop (i.e. first the B frames and then the P frames) based on the inter-frame dependencies as shown in Figure 3.1, we can ensure that all frames that reach the client can be decompressed.

For quality scaling, we use a re-quantization filter. The re-quantizing filter operates on semi-compressed data to reduce bandwidth. It first de-quantizes the DCT-coefficients and then re-quantizes them with a larger quantization step. As quantization is a lossy process the bandwidth reduction by re-quantization is accomplished at the cost of reduction in image quality.

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 this number to four. Table 3.1 shows the different scales and their corresponding frame-rate and bandwidth for 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 bandwidth.

The first level shows the clips at encoded quality and frame rate (30 frames per second). We then have two levels each of temporal and quality scaling. Each temporal scaling method corresponds to a quality scaling method with a similar bit-rate reduction.

We then conducted a user study to evaluate the impact of content-aware scaling on perceptual quality of the video.

3.2 Adaptive Content-Aware Scaling

Having evaluated 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 3.4 shows the architecture of our system.

3.2.1 System Modules

The system consists of 4 distinct modules: *server*, *filter*, *network feedback generator* 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.

- *Filter Module:* The filter module has two kinds of filters: *Temporal Filter* and *Quality Filter*. The temporal filter is a frame dropping filter and does scaling

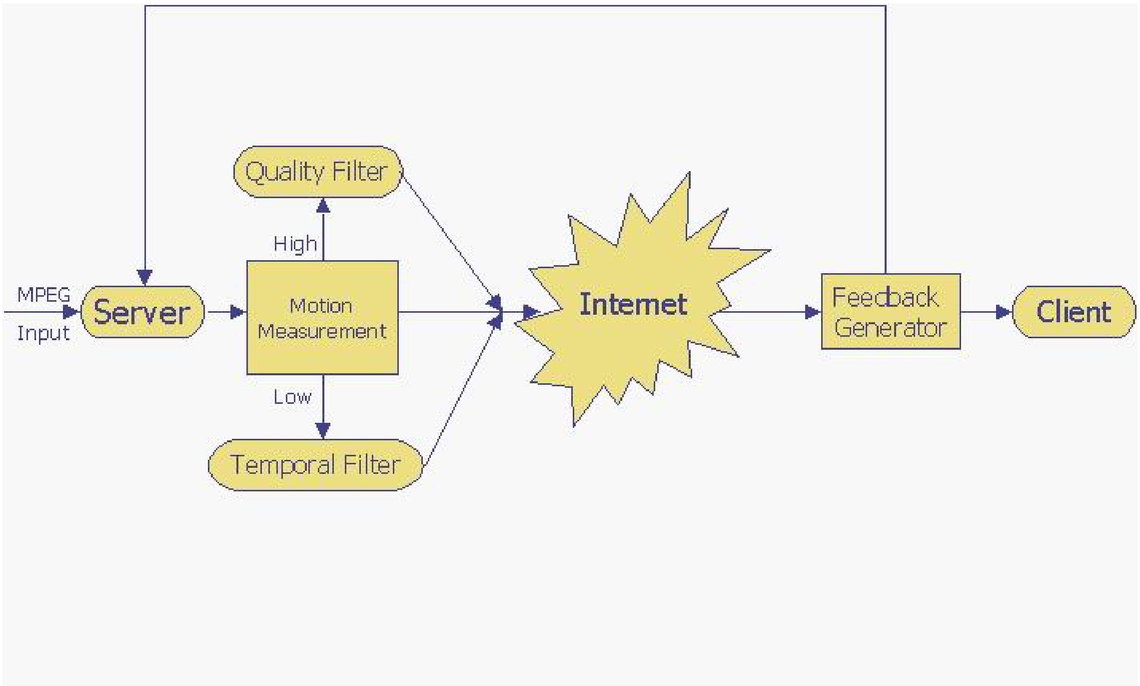

Figure 3.4: Adaptive Content-Aware Scaling System Architecture

in the temporal domain. The quality filter is a re-quantization filter and scales the video stream in the quality domain.

- *Network Feedback Generator:* The network feedback generator module resides on the client side and monitors the congestion in the network by keeping track of the sequence numbers of the packets received. Dropped packets (i.e. packet loss) is taken to be a measure of network congestion. In the event of congestion the feedback module will send control messages to the server indicating a reduction in available bandwidth.

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

Table 3.2: 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

3.2.2 System Functionality

Figure 3.5 shows the sequence of steps that take place in the system. When the server is activated it begins a polling for control messages at a predefined port number. The filter module also polls 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 reads 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 feedback generator on the client side notices the break in sequence and 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 slow motion scene) (Steps 6 through 11).

The system uses 4 distinct scaling levels as shown in Table 3.2.

```
SYSTEM ALGORITHM
(1) ACTIVATE SERVER AND FILTER MODULES
(2) RECEIVE MOVIE REQUEST FROM CLIENT
(3) while not (end_of_file(movie_file)) {
(4)    PARSE REQUESTED FILE AND SEND TO FILTER MODULE
(5)    if (congestion) INVOKE MOTION MEASUREMENT MODULE
(6)       if (highmotion)
(7)          INVOKE QUALITY FILTER
(8)          SEND QUALITY SCALED FRAMES
(9)       else
(10)         INVOKE TEMPORAL FILTER
(11)         SEND TEMPORALLY SCALED FRAMES
(12)   else
(13)      SEND FULL QUALITY FRAMES
(14) }end of while
```

Figure 3.5: Server Algorithm

Chapter 4

Experiments

We conducted two user studies in order to verify the effectiveness of our adaptive scaling system. From the first user study we evaluate the potential benefits of content-aware scaling on perceptual quality of video streams that have consistent motion characteristics. We conducted the second user study to evaluate the potential improvement in perceptual quality of our adaptive content-aware scaling system for streams having variation in their motion characteristics and for different network bandwidth fluctuation rates.

4.1 Experimental Setup

Both the user studies were conducted on identical computer systems in the Fossil lab[1]. In each case three systems were used for the study. The systems had Pentium III, 600 MHz processors with 128 MB of memory. The operating system used was Linux build 2.2.14. The video clips were present on the local hard drives of each of the systems so that network conditions did not influence the video quality. A small graphical interface written in Tcl/Tk (shown in Figure 4.1) was used to record the

[1] http://fossil.wpi.edu

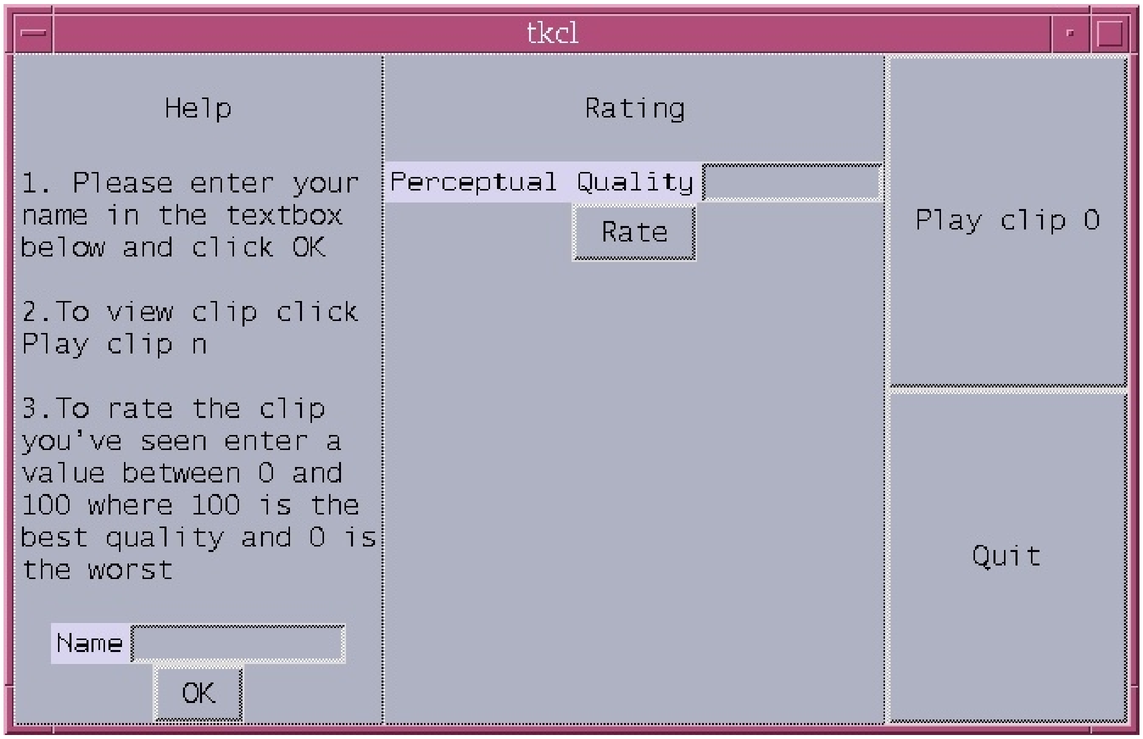

Figure 4.1: Tcl/Tk Interface for users to record Perceptual Quality ratings

responses from the users for both the studies.

4.2 Content-Aware Scaling (User Study 1)

We encoded 18 MPEG video clips from a cross-section of television programming. All the clips were approximately 10 seconds in duration and did not have scene changes in order to have consistent motion characteristics. Using our measure of motion described in Section 3.1, 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 3.1): 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-

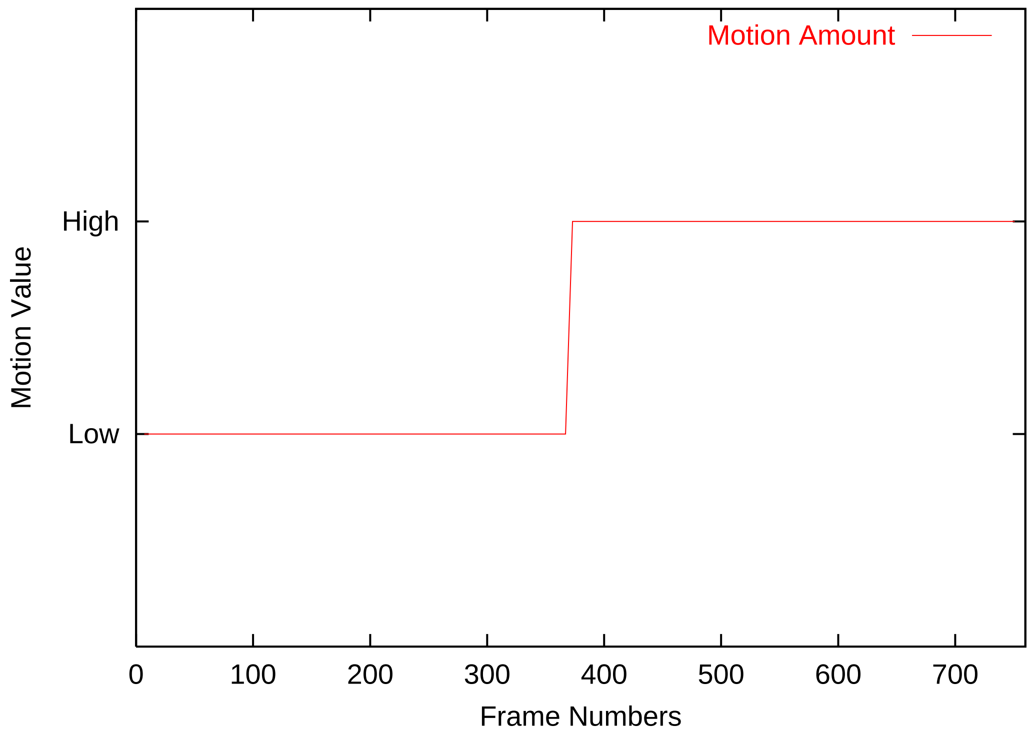

Figure 4.2: Motion Characteristics of Clip 5 (computed motion values on y-axis)

quantization factor set to 31 (quality scaling, level 2).

To evaluate the perceptual quality of the clips, the users 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 given by all users with a 90% confidence interval.

4.3 Adaptive Content-Aware Scaling (User Study 2)

In order to evaluate the benefits of adaptive content-aware scaling on perceptual quality of video, we needed movie clips that had pronounced changes in motion characteristics so that, within the same clip, there is a need to scale the stream

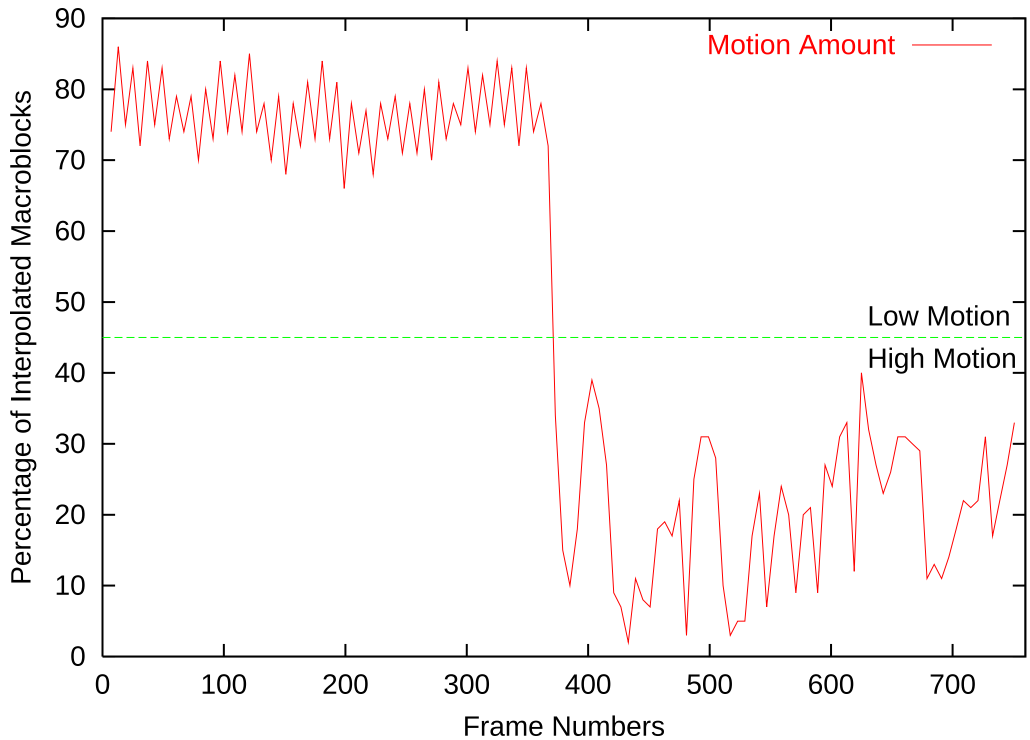

Figure 4.3: Motion Characteristics of Clip 5 (interpolated macroblock percentages on y-axis)

using different scaling techniques (i.e. temporal scaling for parts with slow motion and quality scaling for the parts with high motion).

We encoded 2 clips with varied motion characteristics. Each of the clips was approximately 25 seconds in duration and had one scene change where the transition from low motion to high motion takes place. Figure 4.2 shows the motion characteristics for clip 5 using our measure of motion and Figure 4.3 shows the actual interpolated macroblock percentage values for the clip. The first graph shows frame numbers on the X-axis and the computed motion values on the Y-axis. The second graph shows frame numbers on the X-axis but shows the interpolated macroblock percentages on the Y-axis. Figures 4.4 and 4.5 show similar graphs for clip 6. The clips were approximately 25 seconds in duration and had one scene change where the transition between high motion to low motion or vice versa takes place. Clip

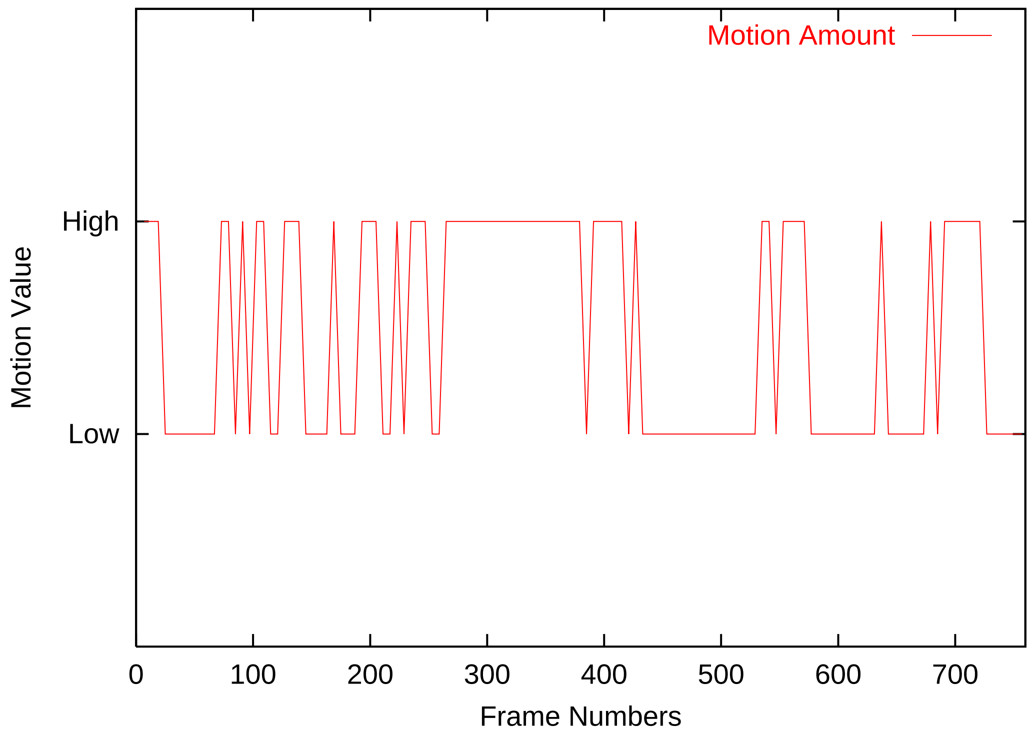

Figure 4.4: Motion Characteristics of Clip 6 (computed motion values on y-axis)

5 shows a scene from a talk show (low motion) followed by a car commercial (high motion). The motion values in this clip are fairly consistent as seen in the graph. Clip 6 shows a scene from the television sitcom *Friends* (predominantly low motion) followed by a commercial for an adventure show (predominantly high motion). Unlike clip 5 there is a considerable amount of variation (≈ 25)in the motion values for this clip.

Bandwidth Distribution Function

The adaptive content-aware system responds to congestion in the network by scaling back the bandwidth of the application intelligently. However, running an application over the Internet introduces too many uncontrollable variables for a careful evaluation therefore, we ran our system on a single machine. Being on a single machine,

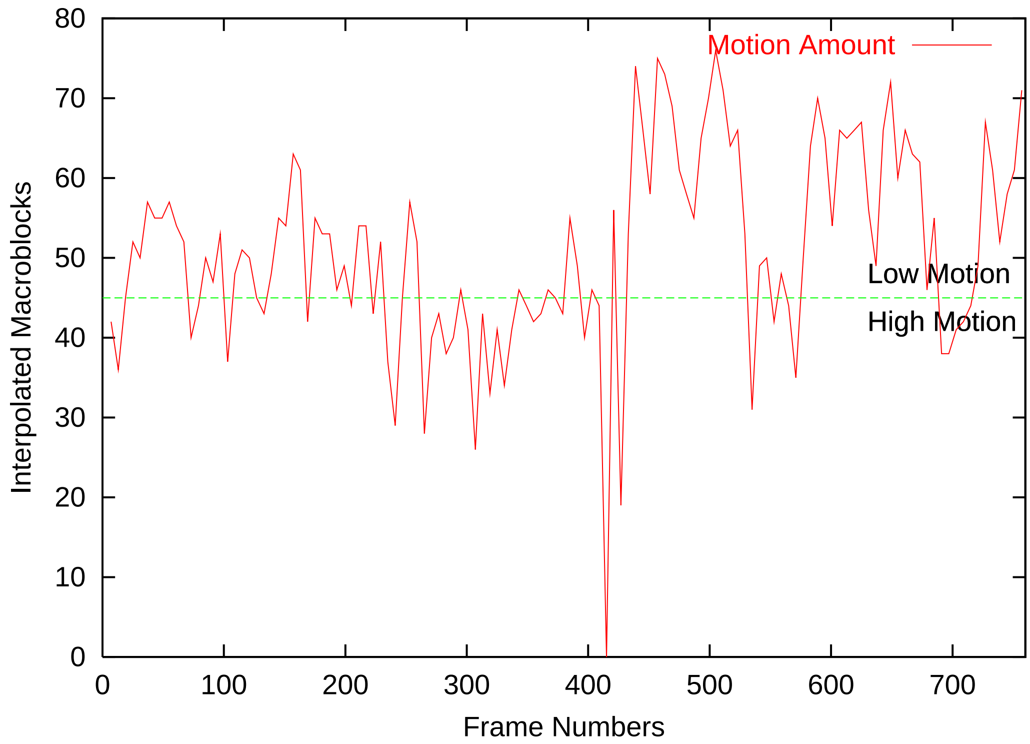

Figure 4.5: Motion Characteristics of Clip 6 (interpolated macroblock percentages on y-axis)

there was no notion of network loss or congestion. We therefore need to simulate these conditions for the application. The simulation uses a normal distribution for the available bandwidth which in turn maps to a normal distribution for the scale levels with an approximate mean scale level of 1 (from Table 3.2) for both temporal and quality scaling.

We ran the server with two different bandwidth distribution functions in order to study the effect that the frequency of variations in video quality has on perceptual quality. In the first version the change in available bandwidth take place every 500 ms and in the second version, changes in bandwidth take place every 2 s. Figures 4.6 and 4.7 show the bandwidth distribution curves for both versions.

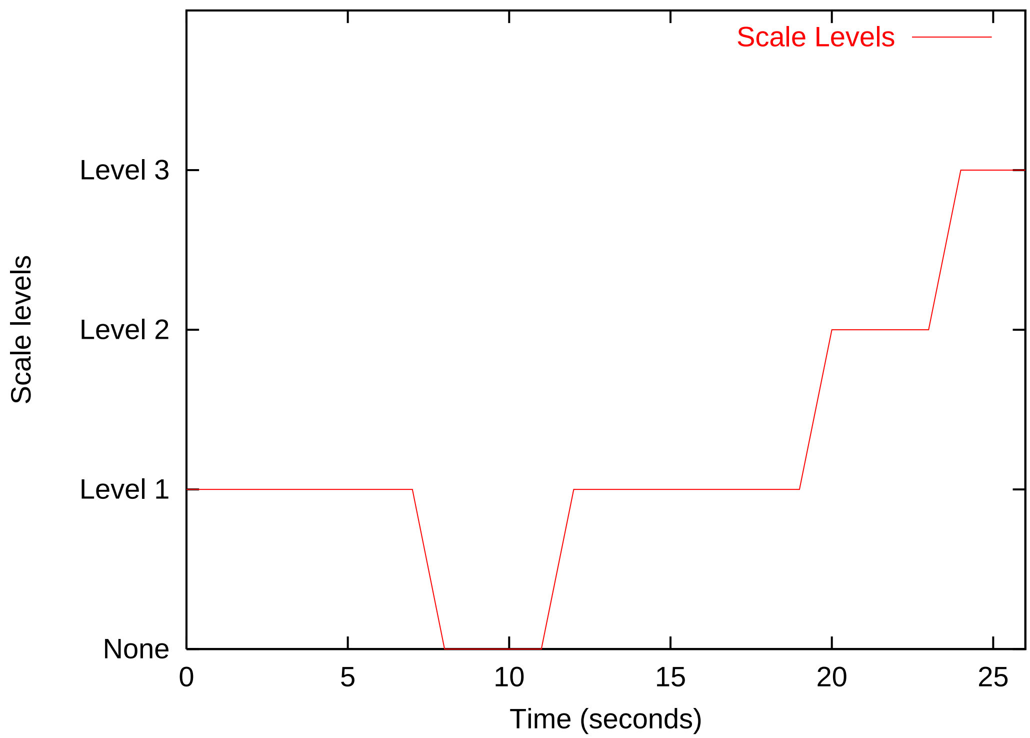

Figure 4.6: Bandwidth Distribution Function (server responds every 2s)

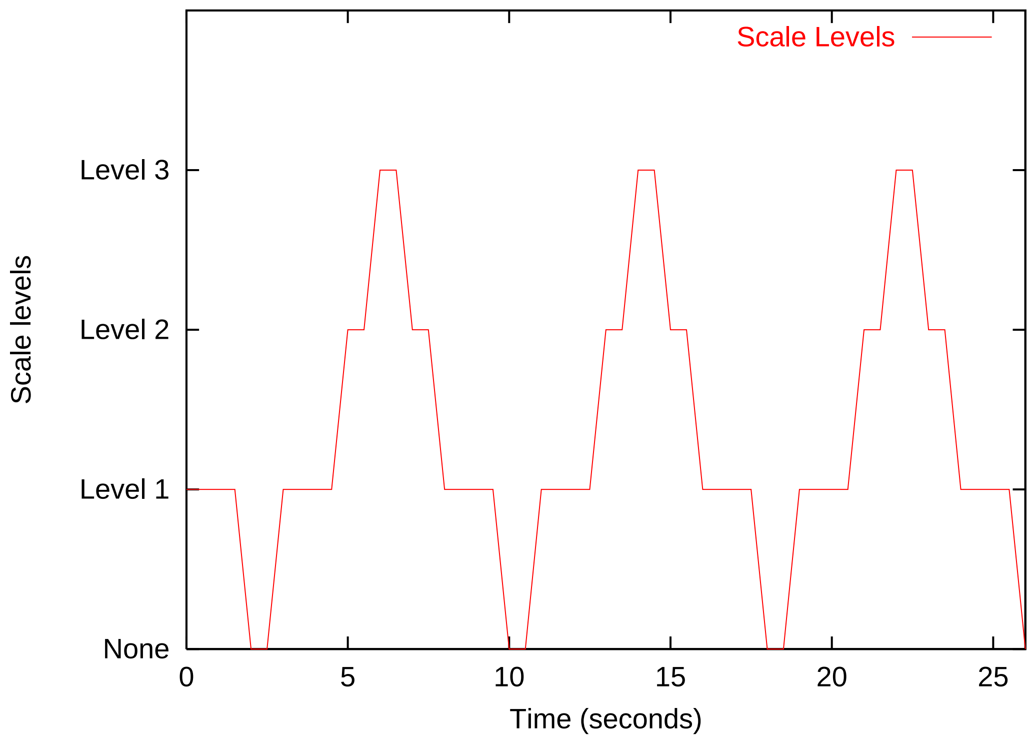

Figure 4.7: Bandwidth Distribution Function (server responds every 500ms)

Chapter 5

Result Analysis

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

5.1 Content-Aware Scaling

In this section we present the results from our first set of experiments (user study 1). For this study we used four 10 second video clips: two having high motion and two having low motion. The clips had no scene changes and their motion characteristics were consistent over the entire duration.

Figure 5.1 shows the graph we obtain when we plot the user perceived quality against the different scaling levels for a low motion clip. This clip 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 more rapidly as the frame rate reduces. We suspect there is a threshold below which users find the perceived qual-

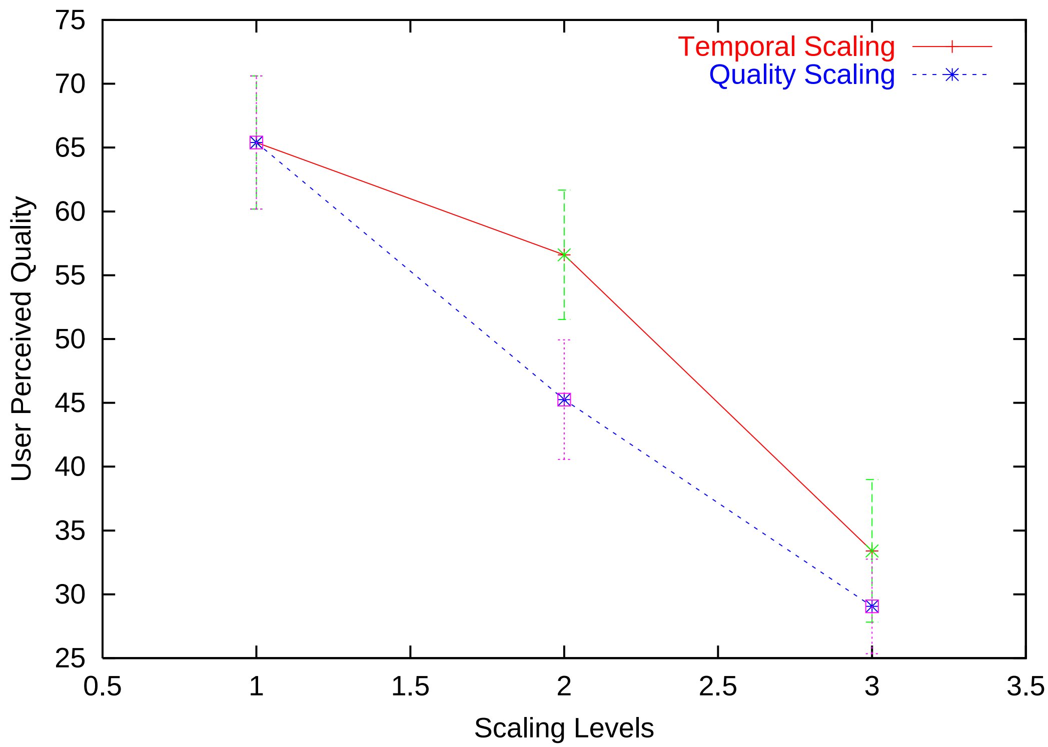

Figure 5.1: Clip 1: Low Motion Clip (70% Interpolated Macroblocks)

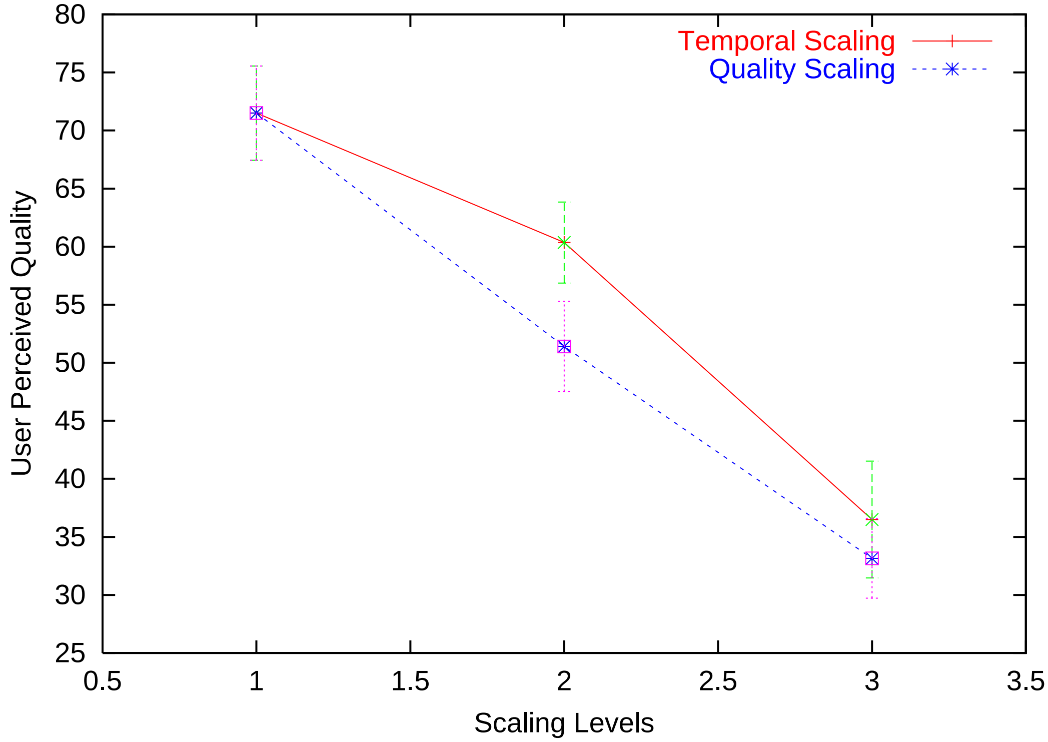

Figure 5.2: Clip 2: Low Motion Clip (57% Interpolated Macroblocks)

ity unacceptable, and when the frame rate drops below this threshold perception of smooth movement is lost. We expect this number to be between 4 to 8 frames per second. The determination of this threshold we leave as future work.

Figure 5.2 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. This clip shows a character from the popular television sitcom "Friends" as she talks on the phone while walking across a room. Here again temporal scaling does consistently better than quality scaling and the user perceived quality drops sharply for the low frame rate of 5 frames per second.

Figure 5.3 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

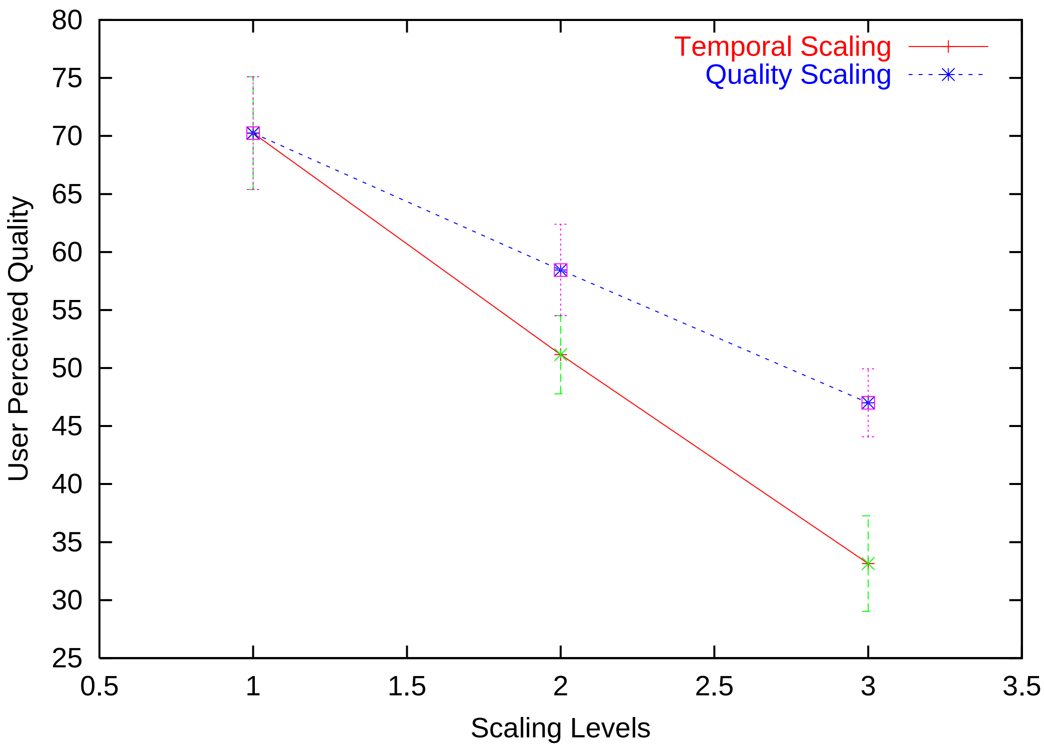

Figure 5.3: Clip 3: High Motion Clip (27% Interpolated Macroblocks)

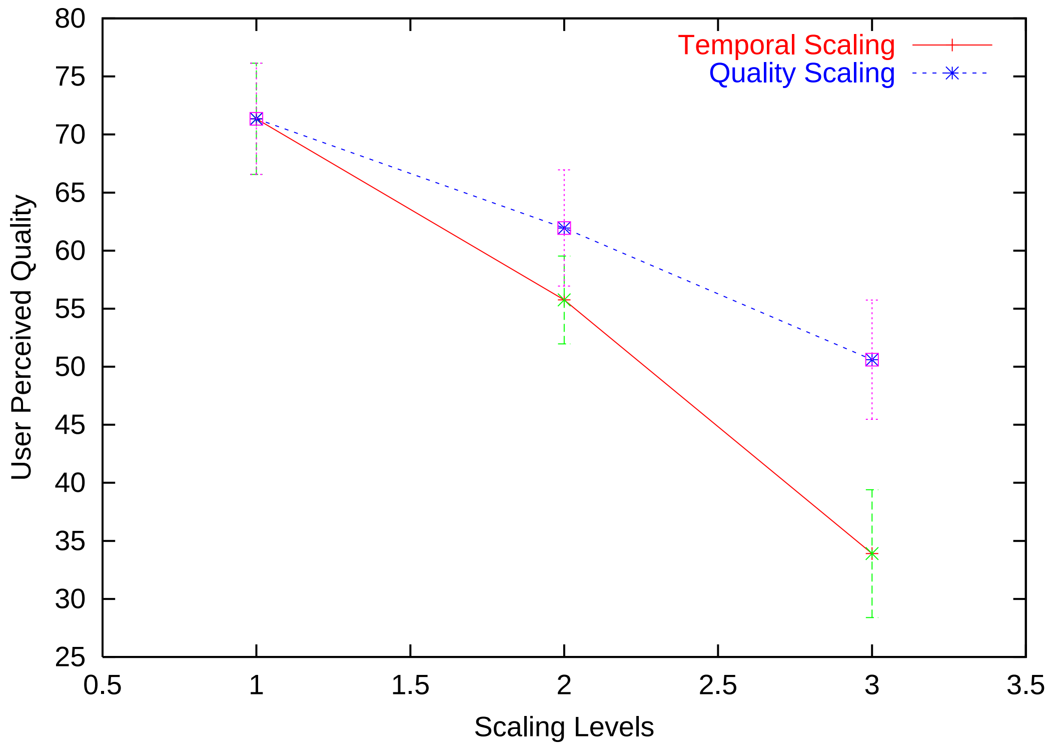

Figure 5.4: Clip 4: High Motion Clip (20% Interpolated Macroblocks)

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.

We obtain a similar graph in Figure 5.4 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.

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

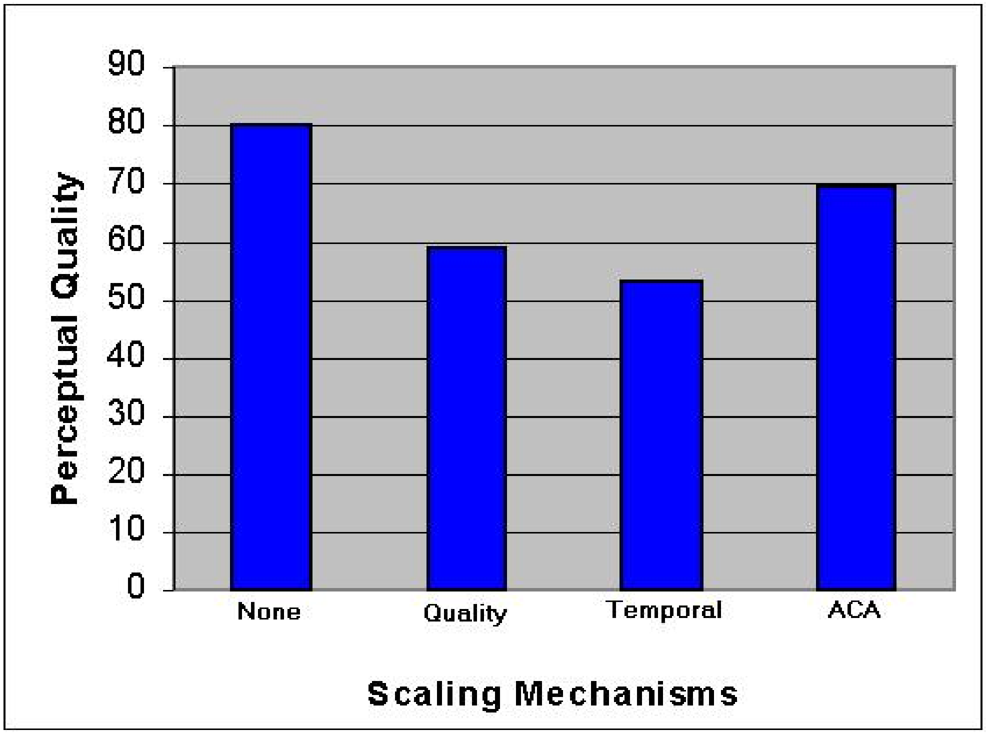

Figure 5.5: Clip 5- Bandwidth changes every 2s

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). The motion values in this clip are fairly consistent as seen in Figure 4.2. Clip 6 shows a scene from the television sitcom *Friends* (predominantly low motion) followed by a commercial for an adventure show (predominantly high motion). Unlike clip 5 there is a considerable amount of variation in the motion values for this clip as seen in Figure 4.4.

Figures 5.5 through 5.8 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 at *None* shows the average perceptual quality value for the clip at full quality without any scaling. The column

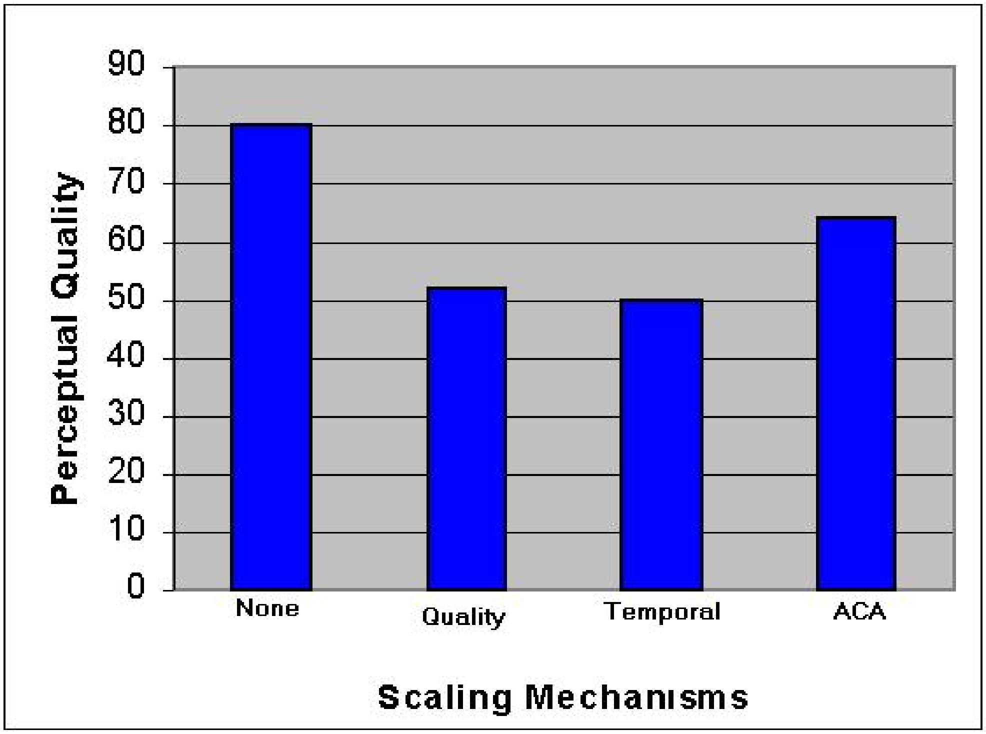

Figure 5.6: Clip 5- Bandwidth changes every 500ms

at *Quality* shows the average perceptual quality when the clip is quality scaled. The column at *temporal* shows the average perceptual quality when the clip is temporally scaled and the column at ACA shows the perceptual quality when adaptive content-aware scaling is done on the clip.

Figure 5.5 shows the graph obtained when 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 5.6 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 adaptive content-aware scaling is done compared to the case where the stream is scaled without regard

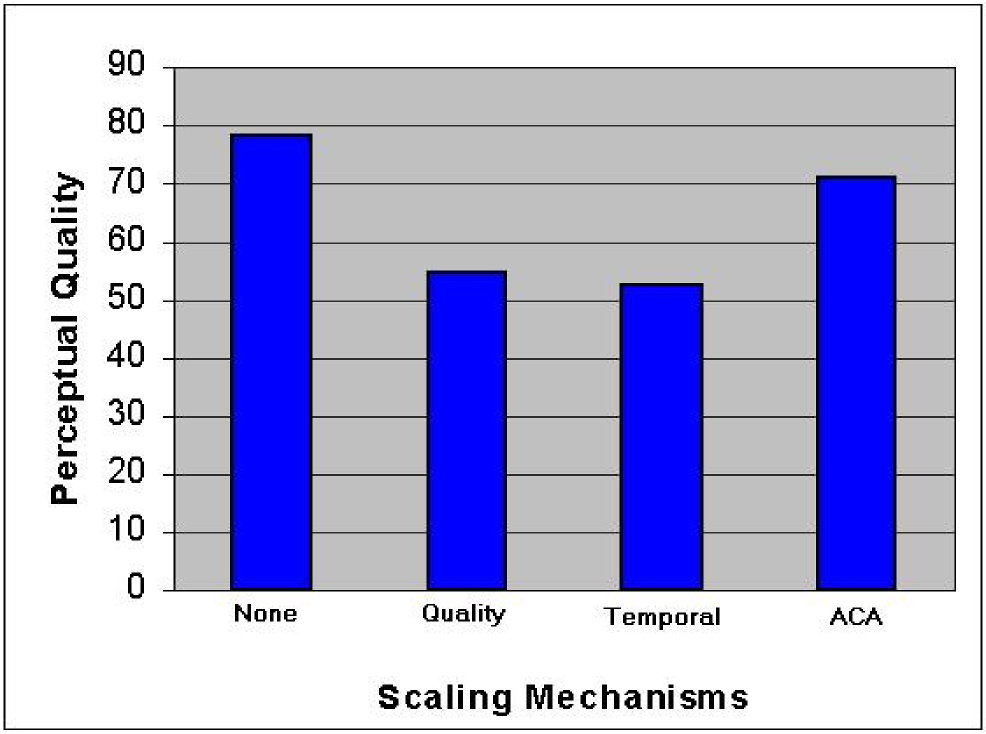

Figure 5.7: Clip 6- Bandwidth changes every 2s

to the content of the stream. The improvement is almost as high as 30% both when bandwidth changes every 2s and when the bandwidth changes every 500ms.

For clip 6 we find that there is an appreciable improvement in the perceptual quality when the available bandwidth changes every 2s 5.7. 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%]. But the improvement is not as high when the bandwidth changes every 500ms 5.8. 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 more

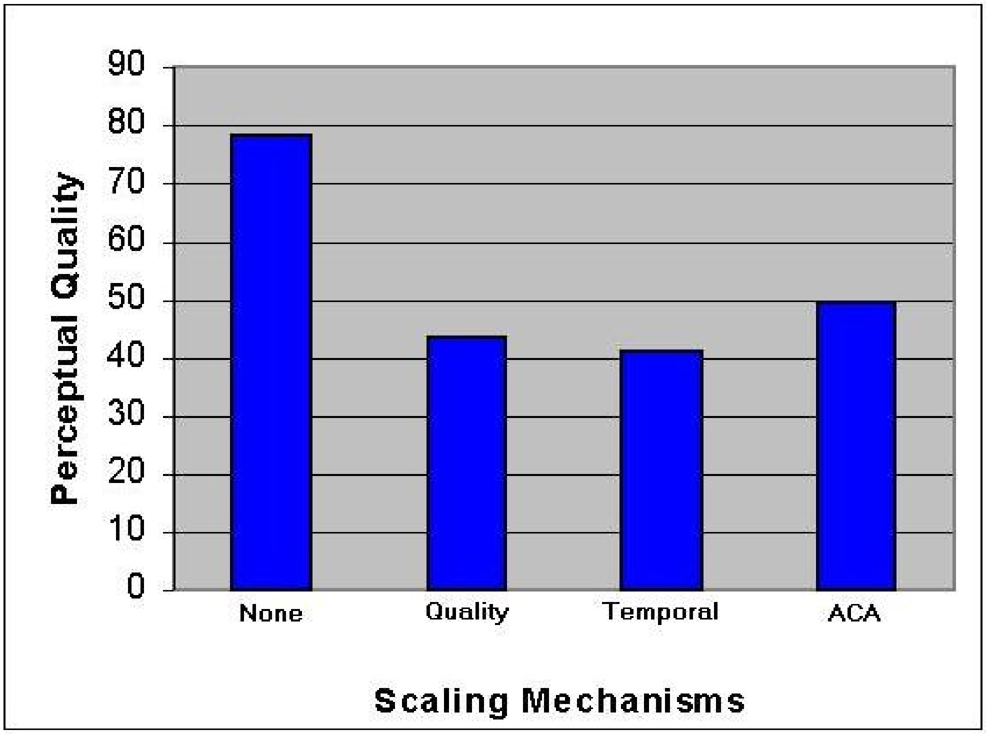

Figure 5.8: Clip 6- Bandwidth changes every 500ms

poorly for the second case.

Chapter 6

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, suggestin 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 applications.

Chapter 7

Conclusions

In this thesis we have presented an application level solution to the problem of congestion due to unresponsive video streams on the Internet. The numerous dependencies between frames in a video stream mean that losing packets from one frame might result in many other frames being rendered useless. In fact, studies have shown that a 3% raw loss rate in an MPEG bit-stream leads to an effective loss rate of 30% at the application level [BG98]. Introducing responsiveness at the application layer can reduce the need for random dropping of packets due to congestion at the routers.

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 the amount of motion in a video stream must be considered when choosing a scaling mechanism for a video stream. For instance, if 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 dropped 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 have 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 varying 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 found 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%.

In summary, the contributions of this thesis include:

- Developed an application level solution to the problem of congestion due to unresponsive video streams on the Internet

- Developed a mechanism to quantify the amount of motion in a video stream

- Showed that content-aware scaling can improve the user perceived quality of video by as much as 50%

- Developed a system to do adaptive content-aware scaling on video streams

- Showed that the improvement in user perceived quality for clips with varying amounts of motion when scaled when scaled adaptively can be as much as 30%

Bibliography

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. Technical Report RFC-2475, Internet RFC, December 1998.

[BCCL99] 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.

[BG98] J. Boyce and R. Gaglianello. Packet loss effects on mpeg video sent over the public internet. In *Proceedings of ACM Multimedia '98*, pages 181–190, Bristol, U.K., September 1998.

[CC00] J. Chung and M. Claypool. Better-Behaved, Better-Performing Multimedia Networking. In *Proceedings of SCS Euromedia Conference (COMTEC)*, May 2000.

[CE99] 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.

[CF98] D. Clark and W. Fang. Explicit allocation of best effort packet delivery service. *IEEE/ACM Transactions on Networking*, 6(4):362–373, August 1998.

[CT99] 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.

[FHPW00] 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.

[FJ93] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, August 1993.

[HHSG99] 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.

[JE98] 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.

[KKSH01] 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.

[LeG91] D. LeGall. MPEG- A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):47–58, April 1991.

[LM97] D. Lin and R. Morris. Dynamics of Random Early Detection. In *Proceedings of ACM SIGCOMM Conference*, September 1997.

[MJV96] S. McCanne, V. Jacobsen, and M. Vetterli. Receiver-driven Layered Multicast. In *Proceedings of ACM SIGCOMM Conference*, August 1996.

[MP96] J. Mitchell and W. Pennebaker. *MPEG Video: Compression Standard*. Chapman and Hall, 1996. ISBN 0412087715.

[MVJ97] 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.

[MWMM00] 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 Circuits/Systems, Computers and Communications (ITC-CSCC 2000)*, July 2000.

[RHE98] R. Rejaie, M. Handley, and D. Estrin. Architectural Considerations for Playback of Quality Adaptive Video over the Internet. Technical Report 98-681, CS Department, University of Southern California, November 1998.

[RHE99] 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.

[SKK00] 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.

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

[WKC+97] 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.

[YGH96] 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.

[YGHS96] 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.