Adaptive Content-Aware Scaling for Improved Video Streaming

Avanish Tripathi and Mark Claypool
Department of Computer Science
Worcester Polytechnic Institute
100 Institute Road,
Worcester, MA-01609, USA
{avanish,claypool}@cs.wpi.edu

ABSTRACT

Streaming video 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. 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. User studies show that our content-aware scaling can improve perceived video quality by as much as 50%.

1. INTRODUCTION

In times of network congestion, the dropping of frames by a router may seriously degrade multimedia quality since the encoding mechanisms for multimedia generally bring in numerous dependencies between frames [4]. For instance, in MPEG encoding, dropping an independently encoded frame will result in the following dependent frames being rendered useless since they cannot be displayed and would be better off being dropped also rather that occupying unnecessary bandwidth. A multimedia application that is aware of these data dependencies can drop the frames that are the least important much more efficiently than can the router [2]. Such application specific data rate reduction is called *media scaling*.

Media scaling techniques for video can be broadly categorized as follows [1]:

- *Spatial scaling:* In spatial scaling, the size of the frames is reduced by encoding 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.
- *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.

It has been shown that the content of the stream can be an important factor in influencing the choice of the preferred scaling technique (i.e. temporal, spatial or quality) [1]. For instance, if a movie scene has a lot of motion and had to be scaled then it would look better if all the frames were played out albeit with lower quality. That would imply the use of either quality or spatial scaling mechanisms. On the other hand, if a movie scene has little motion and needed to be scaled it would look better if a few frames were dropped but the frames that were shown were of high quality. Such a system has been suggested in [3] but the quantitative benefits to multimedia quality for the users has yet to be determined. Other techniques for multimedia scaling have been proposed, 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.

2. 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 a 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. We concentrate on video only since an audio stream takes much less bandwidth than a video stream, and, being more important than the video stream, is typically not scaled.

In the next three subsections we describe the motion measurement, the filtering mechanism and describe the functionality of the full system, respectively.

2.1 Motion Measurement

We have used MPEG video streams 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 [4]. Motion-compensated prediction assumes the current picture

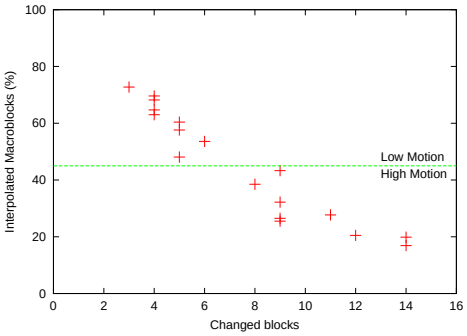

Figure 1: Motion Measurement

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

A typical MPEG stream contains three types of frames: Intra-encoded frames(I), Predicted frames (P) and Interpolated frames (B-for Bidirectional prediction). Each frame is further decomposed into 16x16 blocks called macroblocks, the basic motion-compensation unit. 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[1], each 10 seconds long and containing no scene changes. The frame size was 320x240 with a GOP of 10 frames (IPBBPBBPBB). 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*, an MPEG analysis tool. Figure 1 shows the graph obtained when we plot the percentage of interpolated macroblocks against the number of blocks in which changes were observed when viewing the video clips. The x-axis shows the number of blocks that were observed to change during the movie clip and the y-axis shows the percentage of interpolated macroblocks for the corresponding clip. 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 seems to provide information on visual motion for making decisions regarding scaling policies. Also, the motion measurement and scaling in our system are in two different modules, so our measure of motion could be replaced with an alternate measure of motion, if the new measure was found to be more effective.

[1] All video clips used in this study can be downloaded from http://perform.wpi.edu/downloads

Table 1: Scale Levels for User Study 1

Type	Level	Method	Fps	Bwidth(%)
None	N/A	N/A	30	100
Temporal	1	No B	13	70
Temporal	2	No P or B	5	11
Quality	1	Q = 7	30	65
Quality	2	Q = 31	30	10

Table 2: Scale Levels for User Study 2

Type	Level	Method	Fps	Bwidth(%)
None	N/A	N/A	30	100
Temporal	1	No alternate B	21	85
Temporal	2	No B	13	70
Temporal	3	No P or B	5	11
Quality	1	Q = 4	30	85
Quality	2	Q = 7	30	65
Quality	3	Q = 31	30	10

For our system, we categorize the sequence of frames into two categories, low motion and 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. Based on pilot studies, we compute the motion value for every 4 frames served. Further evaluation of our measure of motion we leave as future work.

2.2 Filtering Mechanisms

We extend the filtering system in [5] to integrate it with our 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. For quality scaling, we use the re-quantization filter. It operates on semi-compressed data, i.e. it first de-quantizes the DCT-coefficients and then re-quantizes them with a larger quantization step. As quantization is a lossy process the bit-rate reduction results in a lower quality image.

Table 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. For the second set of experiments (user study 2) we increase th number of scale levels to four.

2.3 Adaptive Content-Aware Scaling System

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 2 shows the sequence of steps that take place in the system. When the server is activated it waits for a connection on a predefined port number. The filter module also listens 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 module on the client side 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 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 2.

```
(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 2: Server Algorithm

3. EXPERIMENTS

We conducted two user studies in order to evaluate the effectiveness of our adaptive 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 having variation in their motion characteristics and for different network bandwidth fluctuation rates. Both user studies were conducted on Intel P3 600 MHz processor systems with 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. The 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 did not have scene changes 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 1): full quality; no B-frames (temporal scaling, level 1); no B-frames or P-frames (temporal scaling, level 2); re-quantization factor set to 7 (quality scaling, level 1); and re-quantization factor set to 31 (quality scaling, level 2).

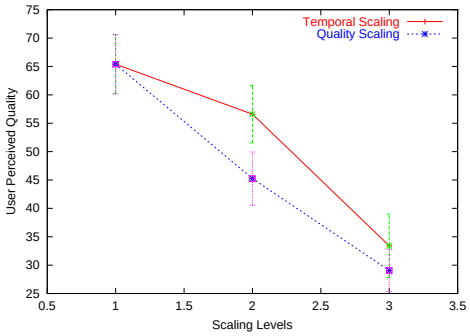

Figure 3: Low Motion Clip (70% Interpolated Macroblocks)

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 scene and the available bandwidth the system automatically selected the most appropriate scaling technique.

4. RESULT ANALYSIS

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

4.1 Content Aware Scaling

Figure 3 shows the graph we obtain when we plot the user perceived quality against the different scaling levels for a low motion clip of four men talking at a bar. 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 quality unacceptable, and when the frame rate drops below this threshold smooth movement is lost. We expect this number to be between 4 to 8 frames per second, and we are currently working on more fine grained scaling levels to accurately determine this frame rate.

Figure 4 shows the graph that we obtain for a high motion clip of a man riding a horse as he tries to catch a bull. It has 27% interpolated macroblocks on an average over the whole clip. As expected, we observe that quality scaling performs consistently better than temporal scaling. We also observe that the drop in user perceived quality for temporal scaling

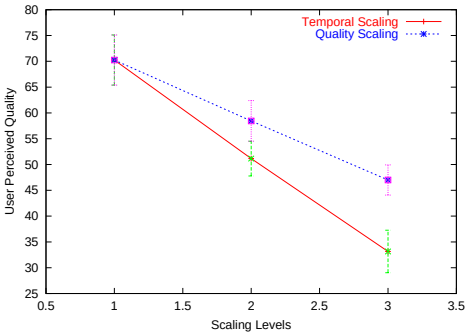

Figure 4: High Motion Clip (27% Interpolated Macroblocks)

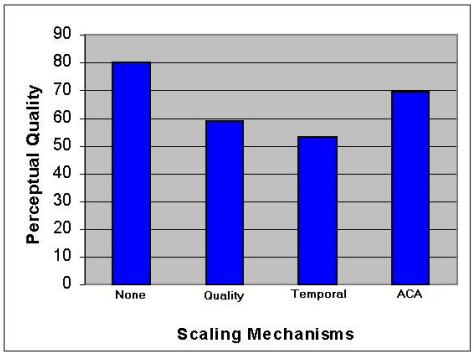

Figure 5: Low-High Motion Clip with Bandwidth Changes Every 2s

level 2 is not as pronounced as in the previous graph 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.

4.2 Adaptive Content-Aware Scaling

Figure 5 shows the graphs we obtain when we plot the perceived quality of the Low-High motion clip (a scene from a talk show (low motion) followed by a car commercial (high motion)) against different scaling mechanisms for varying bandwidths. In 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 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 the clip is adaptive content-aware scaled.

Figure 5 shows the graph obtained when bandwidth changes every 2 seconds for the clip. 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%]. There is an almost 30% 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.

We found similar results for a low-to-high motion clip and for bandwidth changes every 500ms but do not show the results here due to space constraints.

5. CONCLUSIONS

In this paper we have presented an application level solution to adapt to reduced bandwidth in the event of network congestion. 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 find that using content-aware scaling can improve user perceived quality by as much 50% for clips that have consistent motion characteristics over the entire duration of the clip.

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. Also, 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.

6. REFERENCES

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

[2] 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.

[3] 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.

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

[5] 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.