Some Misconceptions about Temporal Constraints of MPEG-2 Video Decoding

Damir Isović, Gerhard Fohler
Department of Computer Engineering
Mälardalen University, Västerås, Sweden
{damir.isovic,gerhard.fohler}@mdh.se

Liesbeth Steffens
Information Processing Architectures
Philips Research The Netherlands, Eindhoven
liesbeth.steffens@philips.com

Abstract

In this paper, we present results from a study of realistic MPEG-2 video streams to analyze the validity of common assumptions for software decoding. Our analysis identifies a number of misconceptions.

We present actual demands based on quality considerations and derive timing constraints for their decoding. We show that standard, fixed timing constraints are restrictive and flexible ones are better suited for MPEG-2 software decoding.

1 Introduction

The Moving Picture Experts Group (MPEG) standard for coded representation of digital audio and video, is used in a wide range of applications. In particular MPEG-2 has become the coding standard for digital video streams in consumer content and devices, such as DVD movies and digital television set top boxes for Digital Video Broadcasting (DVB). A number of algorithms has been presented for efficient decoding and transmission, mostly based on buffering and rate adjustment based on average assumptions. These provide acceptable quality for applications such as video transmissions over the Internet, when drops in quality, delays, uneven motion or changes in speed are tolerable, and time and space is available for buffering. In high quality consumer terminals, such as home TVs, quality losses of such methods are not acceptable. In fact, producers of such devices have argued to mandate the use of hard real-time methods instead [3].

Video data in MPEG is organized in *Group of Pictures* (GOP), i.e., a sequence of pictures that consist of a number of frames. The three types of frames are I frames (*intra-coded pictures*), P frames (*predicted pictures*), and B frames (*bi-directionally predicted pictures*). Simply speaking, I frames contain full pictures and are independent, P frames build a full picture using a previous I or P frame as reference, and B frames contain incremental changes to a full picture, based on both previous and later frames.

An intuitive conclusion is that I will be the largest frames, followed by P and B frames, and frames have similar sizes within their respective frame type. While true on average, such assumptions do not hold for a considerable number of cases. The analysis of realistic streams, movie DVDs, presented in this paper shows, e.g., a case with 9% GOPs in which P frames have the largest size, and 1% GOPs where the largest frame is a B frame. This corresponds to roughly 8 and 1 minutes in a 90 minute movie. Clearly, such deviations from average cannot be ignored. Algorithms based on average behavior, regarding the variations in frame sizes as small deviations will not provide good quality.

When resources, such as processing power or network bandwidth, are limited, adherence of algorithms to appropriate assumptions becomes even more important for video quality: in these situations, not all the frames of the stream can be completely decoded or transmitted. Naive best-effort decoders will simply run out of time when the period for frame display takes place, incurring either a sudden disturbance in smoothness, as pictures are missing, or a delay of subsequent frames, slowing the motion down. As frame decoding starts and proceeds without knowing about timely completion, it may happen that the resources are fully used, but wasted, as partially decoded frames are generally not useful. In extreme cases, the decoding of a large, important frame might just not make it, therefore being lost and impeding quality, while simply skipping to decode a small preceding frame might have freed the resources for completion, with only slight quality reduction. Frame dropping, however, needs appropriate assumptions to be effective. Dropping the wrong - even small - frame at the wrong time can ruin a whole GOP. We found a number of common assumptions including dropping not more than a certain number out of a total, dropping any B frame, dropping frames in order, to not hold in the general case since the number of frames in a GOP can vary and frames have different importance for the overall picture quality.

A model for MPEG video streams that captures the bitrate variations at multiple time scales has been presented

in [8]. Long term variations are captured by incorporating scene changes, which are noticeable in the fluctuations of I frames. MPEG-2 requires adaptive CPU scheduling, due to Variable Bit-Rate [2], i.e., a CPU scheduling that adapts to the frequently varying workloads at variable bit-rates (VBR). A server based algorithm for integrating multimedia and hard real-time tasks has been presented in [1]. It is based on average values for execution times and interarrival intervals. A method for real-time scheduling and admission control of MPEG-2 streams that fits the need for adaptive CPU scheduling has been presented in [5]. The method is not computationally overloaded, qualifies for continuous reprocessing and guarantees QoS. However, no consideration on making priorities on the B frame level has been done.

In this paper we analyse realistic MPEG streams and match the results with the common assumptions about MPEG. We derive realistic timing contraints for frame decoding and display, and show that standard, fixed timing constraints with constant task parameters are restrictive.

2 Analysis of realistic MPEG streams

I frames are self-contained, meaning that they require no additional information for decoding. P and B frames are not self-contained, i.e., if the reference frames are lost, decoding is impossible. Only B frames are never predicted from each other, only from I or P frames. As a consequence, no other frames depend on B frames. Note that the last B frame in a GOP requires the I frame in the next GOP for decoding and so the GOPs are not truly independent. Independence can be obtained by creating a *closed* GOP which may contain B frames but ends with a P frame.

2.1 Common assumptions about MPEG

We have analyzed a number of realistic MPEG streams to get a more clear picture on which assumption about MPEG are valid. Due to space limitations we report only representative results for selected DVD movies. We have chosen an action movie, a drama, and a cartoon movie. The complete results for all analyzed movies can be found in [7]. An overview of the movies we analyzed is summarized in table 1. N and M refer to the GOP length and distance between reference frames respective, e.g. GOP(12,3) means I-to-I distance is 12, while I-to-P and P-to-P distance is 3. We have also analyzed the relations between frame sizes on the individual GOP basis, see table 2.

Assumption 1: - *I frames are the largest and B frames are the smallest*. This assumption holds on average. In all the movies that we analysed, the average sizes of the I frames were larger than the average sizes of the P frames, and P frames were larger than B frames on average. However, our analysis showed that this assumption is not valid for a significant number of cases. For example, in "Mission Impossible 2" we have a case with 9% GOPs in which P have the largest size, and 1% of B frames , which corresponds roughly to 8 and 1 minutes, resp, in a 90 minute film. Such deviations from average cannot be ignored.

Assumption 2: - *An I frame is always the largest one in a GOP*. This is not true. For example in the movie "Mission Impossible 2" the I frame was not the largest in 14% of the cases the I frame might be the most important one in the GOP from the reconstruction point of view, but it does not necessarily has to be the largest one.

Assumption 3: - *B frames are always the smallest ones in a GOP*. Neither this assumption is true. For example, in "Leaving Las Vegas" a B frame was larger than the I frame in 3% of the cases, and larger than a P frame in even 37% of the cases. This implies that even the assumption that P frames are always larger than B frames is also not valid. (e.g., we found a GOP where the B frame is almost 100 times larger than the I frame, $B \approx 1MB, I \approx 12kB$).

Assumption 4: - *The sequence structure in a GOP is fixed to a specific I,P,B frame pattern*. Not true. In 18% of the GOPs in "Mission Impossible 2" the GOP length was not 12 frames. Not all GOPs consist of the same fixed number of P and B frames following the I frame in a fixed pattern. For instance scene changes or large changes in video content do not occur regularly, and hence the need for I frames in most video sequences is not at regular intervals.

Assumption 5: - *Frame properties for all movies are the same*. Neither this is true. Our analysis showed big variations between frame sizes, GOP pattern and the impact on the overall output video quality depending on the number of dropped frames. Different kinds of video will also effect the perceived quality of the video. For instance, the viewer will perceive jerky motion much easier if we drop frames in an action movie than in a cartoon.

Assumption 6: - *B and P frames are sorted in a GOP according to their sizes in descending order*. Not true. As a matter of fact, our analysis showed that the largest B frames are placed towards the end of the GOP. So, the "best-effort" algorithms will perform badly when skipping the last B frames in the GOP.

Assumption 7: - *All B frames are equally important*. Not true. B sizes vary a lot. In our analysis we could see that e.g. in "Leaving Las Vegas" almost 90% of the B frames is in a pretty large interval between 6000 and 300000 bits. So, if we drop a large B frame, the entire GOP could be ruined. On the other hand, more bits does not necessarily mean better quality. That is because motion vectors give the highest compression ratio, but are smallest. So, a B frame with a lot of motion vectors would have less data than some other frame with more raw picture information, but still give

Movie title	GOP	Nr of frames	I frames			P frames			B frames		
			min	max	average	min	max	average	min	max	average
Mission Impossible 2	(12,3)	179412	256	1782128	516763	16	766696	226720	16	1075100	145051
Leaving Las Vegas	(12,3)	173054	136	1469848	140329	32	1009832	76835	32	636416	45746
Chicken Run	(12,3)	121406	7424	1121216	674549	272	1097336	255551	264	891240	115185

Table 1. Frame size statistics for selected analyzed MPEG streams (in bits)

Movie title	Open GOPs	Closed GOPs	GOPs with same length	Number of GOPs where					
				I largest	P largest	B largest	$P > I$	$B > I$	$B > P$
Mission Impossible 2	83%	17%	82%	90%	9%	1%	9%	5%	39%
Leaving Las Vegas	98%	2%	92%	94%	5%	1%	6%	3%	37%
Chicken Run	99%	1%	98%	92%	7%	1%	8%	1%	12%

Table 2. GOP statistics

better output quality when decoded. All this implies that dropping of B frames should be performed carefully.

Assumption 8: - *Frame sizes vary with minor deviations from the average value*. Not true. In "Mission Impossible 2" frame sizes vary greatly around an average of 227000 bits. The interval between 153000 and 306000 bits (0.5 and 1.5 of average) holds only some 60% of frames.

Assumption 9: - *Decoding time depends on the frame size and it is linear*. While some results on execution times for special kinds of frames have been presented, e.g., [4], a (linear) relationship between frame size and decoding time cannot be assumed in the general case. The execution time of decoding tasks depends also on the compression used; B frames require resource-intensive compression techniques such as Motion Compensation and Motion Estimation - which may result in extensive decoding work from small frame sizes; at some points in the stream, the transformation matrices may have to be recalculated. We are currently investigating this assumption.

3 Flexible Timing Constraints for MPEG-2

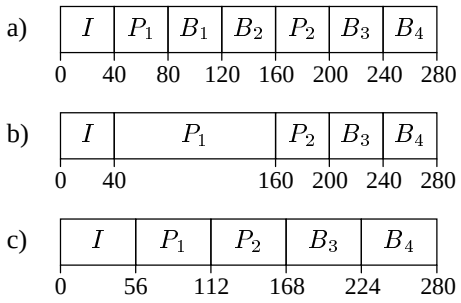
After performing analysis of various MPEG streams, we are now ready to derive the timing constraints needed for the decoding tasks and the display task. In our model, different frame types are decoded by different tasks, with different number of invocations for each GOP. We have three decoding tasks, D^I, D^P and D^B each for respective frame type.

3.1 Periodicity

Task D^I will be invoked once per GOP, D^P and D^B are invoked once for each P or B frame. However, those three tasks does not have to be strictly periodic. They are periodic under the conditions that we do not drop frames and the GOP length is fixed. But our analysis showed that not all GOPs in a stream have uniform length. Besides, we might need to drop frames in resource limited systems. Hence, the decoding tasks will have repetitive behaviour, but not strictly periodic. For instance, if we skip a P frame, then D^P does not need to be invoked for that particular frame. Instead of calling them periodic or aperiodic, we rather refer to them as tasks with *flexible timing constraints*.

3.2 Start times and deadlines

Each frame must be decoded and displayed within the specified "frame-per-second (fps)"-rate, *fps*, so the common way to assign a deadline for decoding and displaying of a frame f_i is $dl(f_i) = f_i * (1/fps)$, e.g., if fps=25, then the deadline for the fifth frame will be 5*1/25 =0.2s= 200ms. However, we do not think this is the best way to assign deadlines. Assume for example a GOP as in case a) below, with the deadlines.



What happens with deadlines if we must skip some frames? For instance, in order to decode the first two B frames, the P_1 frame must be decoded and displayed before time t=80 ms, counted from the GOP arrival. But what if we must drop both B_1 and B_2 due to the high system load? Then there is no point in having such a tight deadline for the P_1 frame, since neither B_1 nor B_2 will be decoded, i.e., we have time until t=160 ms to decode and display the P_1 frame, as de-

scribed in case b). Another way to gain from the fact that B_1 and B_2 must be dropped is to distribute resources needed for their decoding (80 ms) to all of the frames in the GOP, not only their predecessor. So, instead of extending the deadline of P_1 by 80 ms, we rather give 80/5=16 ms to each of the remaining frames, as depicted in case c) of the figure above, i.e, we relax the deadlines of remaining frames. This will not only provide for the acceptance of more frames (due to the relaxed deadlines), but also it will make the video more smooth. Similar reasoning is valid for the start times; if a frame is decoded earlier, we start with the next one right away. Hence, even start times and deadlines for frame decoding tasks will be flexible, following the above proposed model of tasks with flexible timing constraints.

Display task Even the display task will have flexible timing constraints. If we do not drop any frames, the period of the display task will be equal to the fps-rate, since we need to display all the frames. However, if we drop some of the frames, then there is no need to invoke the display task that often, e.g, if the GOP length is 12 and we drop 5 frames, then we will run the display task only 7 times, not 12.

4 Current work

The analysis presented here focused on the frame level. We are currently investing the impact of the sub frame level, e.g., execution, motion vectors, as well.

Not all the frames are equally important for the overall video quality. Our idea is to set importance values (priorities) to frames before actually trying to guarantee them. We have used the analysis results of realistic MPEG streams to identify a set of criteria that are to be applied to each frame when deciding its importance for the overall picture quality. Each frame in a GOP is assigned unique values, depending on how important a particular frame is for the entire GOP. The importance values are set with respect to the frame type, size, position in the GOP and smoothness of the video (if frames are skipped evenly, then the picture information loss will be more spread, giving smoother video [9]).

We are formulating an algorithm that will select frames according to their importance values, calculated with respect to *all* mentioned criteria applied together (rather than a single criterion). It could, for instance, happen that we skip a frame that is bigger than some other frame in the same GOP, but also less important for the overall picture quality. The algorithm creates an ensemble of decoding tasks for the frames in the GOP, each with timing constraints suited specifically for the particular GOP.

We are also looking into how we can apply the frame selection algorithm on top of an existing scheduling algorithm. As an example, we will show how MPEG streams can be handled with the scheduling algorithm that we pesented in a previous paper [6]. Note however that a variety of other scheduling algorithms with some form of guarantee mechanism can easily be used, since we have separated the assignment of the importance values from the scheduling.

5 Conclusion

In this paper, we presented a study of realistic MPEG-2 video streams and showed a number of misconceptions for software decoding, in particular about relation of frame sizes, equal importance of frames, and variations of sizes.

Using the analysis, we determined realistic flexible timing constraints for MPEG decoding, for use in formulating a quality based frame selection algorithm. Our current work includes extending the study to the sub frame level, e.g., relationship between framesize and execution time, motion vectors, and sub frame decoding. Furthermore, we are formulating a quality based frame selection algorithm to be used in a real-time scheduling framework.

References

[1] L. Abeni and G. C. Buttazzo. Integrating Multimedia Applications in Hard Real-Time Systems. In *Proceedings of the 19th IEEE Real-Time Systems Symposium*, Madrid, Spain, 1998.

[2] V. Baiceanu, C. Cowan, D. McNamee, C. Pu, and J. Walpole. Multimedia Applications Require Adaptive CPU Scheduling. In *Proceedings of the Workshop on Resource Allocation Problems in Multimedia Systems*, Washington, DC, USA, 1996.

[3] R. J. Bril, M. Gabrani, C. Hentschel, G. C. van Loo, and E. F. M. Steffens. QoS for consumer terminals and its support for product families. In *Proceedings of the International Conference on Media Futures*, Florence, Italy, May 2001.

[4] L. O. Burchard and P. Altenbernd. Estimating Decoding Times of MPEG-2 Video Streams. In *Proceedings of International Conference on Image Processing (ICIP 00)*, Vancouver, Canada, September 2000.

[5] M. Ditze and P. Altenbernd. Method for Real-Time Scheduling and Admission Control of MPEG-2 Streams. In *The 7th Australasian Conference on Parallel and Real-Time Systems (PART2000)*, Sydney, Australia, November 2000.

[6] D. Isovic and G. Fohler. Efficient Scheduling of Sporadic, Aperiodic, and Periodic Tasks with Complex Constraints. In *Proceedings of the 21st IEEE Real-Time Systems Symposium*, Orlando, Florida, USA, November 2000.

[7] D. Isovic and G. Fohler. Analysis of MPEG-2 streams. Technical Report at Malardalen Real-Time Research Centre,Vasteras, Sweden, March 2002.

[8] M. Krunz and S. Tripathi. On the characterization of VBR MPEG streams. In *Proceedings of ACM International Conference on Surement and Modeling of Computer Systems (SIGMETRICS 97)*, Seattle, Washington, USA, June 1997.

[9] J. K. Ng, K. R. Leung, W. Wong, V. C. Lee, and C. K. Hui. Quality of Service for MPEG Video in Human Perspective. In *Proceedings of the 8th Conference on Real-Time Computing Systems and Applications (RTCSA 2002)*, Tokyo, Japan, March 2002.