

Quality Variations of Different Priority-based Temporal Video Adaptation Algorithms

Michael Kropfberger, Klaus Leopold, and Hermann Hellwagner

Department of Information Technology, University Klagenfurt, Austria

Email: {mike,klaus,hellwagn}@itec.uni-klu.ac.at

Abstract—If videos are streamed over heterogeneous networks like the Internet, severe bandwidth fluctuations can emerge which hamper seamless transmission to the end user. To overcome this issue, a video’s quality and, as a consequence, its bandwidth requirements can be reduced. Quality reduction in the temporal domain (i.e., frame dropping) turned out to be a promising approach because it is fast and easy to perform. In this paper, we present different approaches for temporal video adaptation and we investigate their performance in terms of the achieved visual quality when applied on several videos. The results show that our QCTVA approach (quality controlled temporal video adaptation), based on PSNR evaluation of frames, yields superior quality.

I. INTRODUCTION

Streaming multimedia data over networks is becoming increasingly important, not only for pure entertainment, but also in areas like distance education or product presentations on the Web. In best effort networks like the Internet, it is often the case that the bandwidth consumption of a video exceeds the available bandwidth of the network link. In such a case, it is not possible to transfer all the data timely and accurately. However, when reducing the quality of the video, the bandwidth consumption decreases and, thus, it may become again possible to fulfill the bandwidth constraints and to properly stream the video to the consumer. There do exist a lot of video adaptation approaches for controlled quality reduction to achieve the aim of bandwidth reduction. Temporal adaptation turned out to be a promising adaptation technique because it is fast and easy to perform. Applying temporal scalability on a video stream means dropping frames before transmission and, thus, immediately decreasing the required bandwidth. The task to not transmit data is easy to perform and, hence, it consumes little computational power. As a consequence, temporal video adaptation can be applied on network nodes like, e.g., routers or multimedia gateways, which have to process a lot of network traffic and which have to follow soft real-time constraints.

Many approaches simply drop arbitrary frames without spending much effort on finding an intelligent dropping behavior. This usually leads to good adaptation results in terms of bandwidth reduction but also to suboptimal video quality. This paper introduces various approaches for temporal video adaptation which consider the visual loss in quality when fulfilling given network bandwidth constraints, but also evaluate their computational needs for preparation and processing.

Furthermore, we present quality measurements of videos that are streamed and temporarily adapted in our network test environment.

II. TEMPORAL VIDEO ADAPTATION

In many state of the art video codecs like, for instance, MPEG-2 and MPEG-4, each video elementary stream consists of a sequence of different frame types which are: I-frames, P-frames, and B-frames [1]. I-frames are independent from any other frames, P-frames are based on prediction from the last reference frame, and B-frames are based on prediction from the previous and the following reference frame. A reference frame might be either an I-frame or a P-frame. No frame type needs B-frames for decoding and thus, they can be dropped at will without interfering with the decoding process of the video.

Semi-intelligent Frame Dropping

If a network node, e.g., a server or media gateway, detects that the video cannot be streamed due to bandwidth constraints, it can reduce the bitrate by dropping B-frames. We call such network nodes *semi-intelligent* because they can at least distinguish between frame types. The easiest approach for bandwidth reduction is to stream as long as bandwidth is available and, if the allowed bandwidth portion is exceeded, the server blocks and drops frames until the bandwidth is available again. A possible frame dropping pattern for this behavior may look like this: `IBBBBPBBBBPBBBBPBBB-P----`, where I, P, and B are the frame types which are transmitted to the client and ‘-’ means that the frame is dropped. The beginning of this frame pattern exhibits no frame loss, but at the end almost every frame is dropped. As the human eye and brain are trying to track smooth motion of objects [2], the visual perception of this dropping behavior is observed as very choppy.

Timely Uniform Distributed Frame Dropping

Based on the awareness that the human eye and brain are trying to track smooth motion, a timely uniform distributed dropping behavior appears to be more reasonable. To achieve timely uniform distributed frame dropping for eg. each even frame number, it is enough to calculate the modulo two function for the incoming B-frame number and accordingly keep or drop the frame. The resulting frame dropping pattern may look like this: `I-B-P-B-P-B-P-B...`

For more fine-grained prioritization, an algorithm was developed, which produces timely uniform distributed priority patterns and stores them in a lookup table. Basically, it implements a recursive depth-search, which is limited to a certain depth. At each depth, left and right traversals are started, which sets ascending priority numbers to the alternating tree halves.

The following example in Table I shows the needed three depths for a pattern size of seven frames, and how the new priorities are uniformly assigned for each depth. In the first depth level, no priorities are assigned to the table of frames yet, which is denoted as double dots (. .). Only the center frame gets the highest number, so this frame will be dropped first, when priority-based adaptation is necessary. After the algorithm has worked down to the third depth level, all frames have assigned priorities, which are perfectly timely uniform distributed.

1:	. . - . . - . . -	7 - . . - . . - . .
2:	. . - 6 - . . -	7 - . . - 5 - . .
3:	4 - 6 - 2 -	7 - 1 - 5 - 3

TABLE I

BUILDING TREE FOR TIMELY UNIFORM DISTRIBUTED PRIORITIZATION
FOR SEVEN FRAMES

Using this table, it is possible to drop frames at every possible priority level. Eg. to drop three timely uniform distributed frames, we just look up the positions of the priority values 1,2,3. Using pattern IBBBBBB, this results in a frame pattern of IB-B-B-.

Frame Prioritization based on Quality Computation

Assume a frame pattern of a video where the first images are almost identical like, e.g., a close-up of a person, followed by a pan shot where a lot of motion can be observed. The semi-intelligent adaptation node might transmit all almost identical images from the close-up scene and start dropping frames when the pan shot starts. This is obviously the worst scenario because frames are dropped when motion can be observed. Applying the timely uniform distribution algorithm on the same scene will give better results because frame drops are uniformly distributed over high- and low-motion scenes. However, it would be even better if almost no frames were transferred in the close-up scene but as many as possible in the pan shot scene.

Exactly this behavior can be achieved when analyzing the video stream with QCTVA (quality controlled temporal video adaptation) [3]. QCTVA offers the qualitatively best temporal scalability by measuring the visual quality of possible frame dropping combinations based on the well-known PSNR value. To evaluate the quality of a frame sequence, the average Peak Signal-to-Noise Ratio (PSNR) is calculated for every frame. The quality of a pattern with dropped frames is calculated by using the last available frame instead of the dropped frame.

Table II demonstrates this behavior where, e.g., frame number 3 was dropped; frame number 2 was replayed instead, which resulted in a quality reduction from 33.984 dB down to 23.984 dB. In this example, the subsequent frames 4 and 5 were also dropped, which resulted in replaying frame 2 even longer. PSNR values dropped from 34.191 dB to 19.191 dB and from 34.432 dB to 14.432 dB, respectively. The quality loss is growing with each frame, since the error is propagated due to continuing motion in the video.

No	Type	PSNR [dB]	No	Type	PSNR [dB]
1	I	35.342	1	I	35.342
2	B	33.993	2	B	33.993
3	B	33.984	2	B	23.984
4	B	34.191	2	B	19.191
5	B	34.032	2	B	14.032
6	P	35.561	6	P	35.561
7	B	34.432	6	P	14.432
8	B	34.331	8	B	34.331
9	B	34.531	9	B	34.531
10	B	34.667	9	B	14.667
11	P	35.123	11	P	35.123
Original quality		34.562	After dropping		26.835

TABLE II

PSNR ESTIMATION IN CASE OF RANDOMLY DROPPED B-FRAMES

To compute the quality of any frame dropping pattern, a *modification lattice* of all frame sequences (with zero, one, or more dropped frames) has to be built. In the lattice, the original frame sequence is the *master pattern* and sequences with frames dropped are referred to as *modifications*. All modifications with the same number of dropped frames are denoted as a *layer*. The modification where no more B-frames are available is called the *base layer*. To get the best dropping behavior, one has to select the node with the maximum quality value on a layer. Due to the construction of the modification lattice, not only the best dropping behavior can be calculated but also the worst and the average modifications which serve as good comparison points.

As the computation of the whole modification lattice is a very time consuming task, heuristics for best first expansion (BFE) and worst first expansion (WFE) were developed which are discussed in detail in [3]. Figure 1 illustrates a modification lattice which was constructed using the BFE heuristic.

Based on the modification lattice, priorities of frames can be derived. I-frames always have the highest priority 1 and P-frames have priority 2 because usually it does not make sense to drop them. Priority 3 is assigned to the B-frame which is dropped at the lowest layer, priority 4 to the B-frame which is dropped at the lowest but one layer and so forth.

In Figure 1, starting from the bottom of the lattice, the priority of the frames in the base layer is 1 for the I-frame and 2 for the P-frame. The priority for the B in I-BP-- is 3 because it is the last B-frame before the base layer. In the pattern I-BPB-,

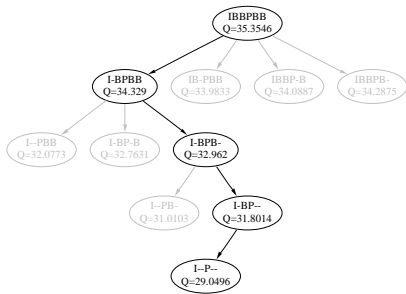


Fig. 1. Example QCTVA lattice with best first expansion (BFE)

No	Type	Prio	Size	FrOffset
0	I-VOP	1	7908	0
1	B-VOP	6	1677	7908
2	B-VOP	3	1579	10585
3	P-VOP	2	2540	12164
4	B-VOP	4	1785	13704
5	B-VOP	5	1538	16489

TABLE III
FRAME PRIORITIZATION BY QCTVA

the fifth frame is new and therefore is assigned priority 4. Analogously, the last B-frame in $I-BPBB$ gets 5 and the second frame in $IBPBB$ gets 6 as its priority. Table II illustrates the frame priorities determined by QCTVA.

III. MEASUREMENTS

For the measurements, we implemented a test environment which is capable of streaming and temporarily adapting videos based on meta information about frame priorities [4]. Furthermore, the test environment features the extended RTP mechanisms immediate feedback [5] and retransmission [6]. The tests were conducted on CIF reference streams defined by the MPEG consortium (see Table IV). All video sequences were encoded using the Microsoft MPEG-4 reference implementation with 30 fps and a fixed frame pattern of 4 B-frames between I- or P-frames, resulting in one-second GOPs as follows: $IBBBBBPBBBBPBBBBPBBBBPBBBBPBBBB$. Note that, depending on the video, more or less motion has to be encoded, even though fixed quantization values for all frames are used. This results in severely different bandwidth requirements for the test sequences to reach about similar average PSNR values.

The following PSNR comparisons consider the QCTVA BFE (best first expansion) as the best achievable result, QCTVA WFE (worst first expansion) as the lowest possible quality, both calculated in the uncompressed domain in a time-consuming video analysis process. Timely uniform distributed frame dropping as an intermediate “prioritization” algorithm has low algorithmic complexity and can be performed in the compressed domain.

Figure 2 shows the quality loss (in terms of PSNR reduction, averaged over an entire video) resulting from temporal adaptation enforced by bandwidth restrictions.

The available network bandwidth was set to 85% and 70% of the average bandwidth required by a video, respectively. Especially for the 85% case, QCTVA BFE is superior since the frames to be kept (or dropped, respectively) are correctly chosen. Yet, the more adaptation is needed (e.g., 70% or lower), the more QCTVA BFE and any other prioritization scheme converge to the base layer of I- and P-frames and therefore deliver more and more similar results.

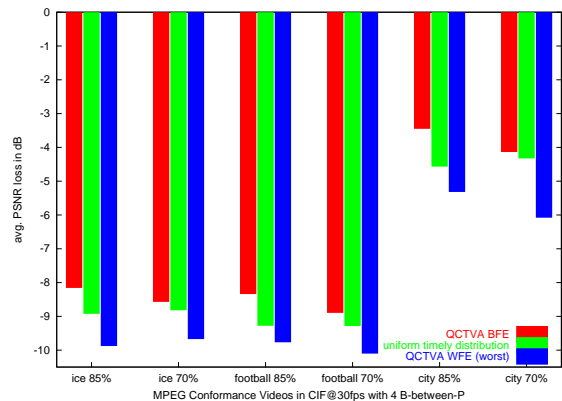


Fig. 2. Quality loss due to frame dropping at 85% and 70% of the required average network bandwidth

For the test sequence *ice* at the 85% bandwidth rate, timely uniform frame dropping marks nearly the middle of QCTVA BFE and QCTVA WFE. The video shows different motion patterns within a frame for each person sliding over the ice. Since PSNR changes are inherent all over the video, but not directly related to frame-per-frame motion, timely uniform distributed frame dropping performs reasonably well, though not perfectly.

For the *football* scene, QCTVA BFE outperforms timely uniform frame dropping even more, since the motion is unevenly distributed over the video in terms of unexpected camera pans. First, there is a still-camera scene on the melee of players. Here, the same rules as for *ice* apply. But then, all of a sudden, a player makes a pass and the ball flies all over the field. The camera tries to follow the ball, which introduces high motion for a short time. QCTVA BFE is capable of optimizing priorities for this fast scene and drops frames at the slow motion scene. Timely uniform distributed dropping discards frames uniformly all over both scenes, which causes unnecessary frame loss also in the fast scene.

For the all-time slow-motion video *city*, the average PSNR loss is low for all three algorithms. Still, since the scene is recorded with a hand camera from within an airplane, there are some bumpy camera moves. Those are better detected and prioritized by QCTVA BFE than by timely uniform dropping.

Name	Frames	Overall PSNR [dB]	kbps 100%	kbps 85%	kbps 70%	Description
ice	236	32.53	243	207	170	About 15 people are ice skating, each person moves at different speed
football	256	29.53	591	502	414	Football game with still camera on game play, then a jerky high motion camera pan follows a thrown ball
city	296	29.49	241	205	169	Smooth and slow camera pan over city skyline, recorded from an airplane

TABLE IV
MPEG-4 REFERENCE STREAMS USED FOR B-FRAME PRIORITIZATION COMPARISON

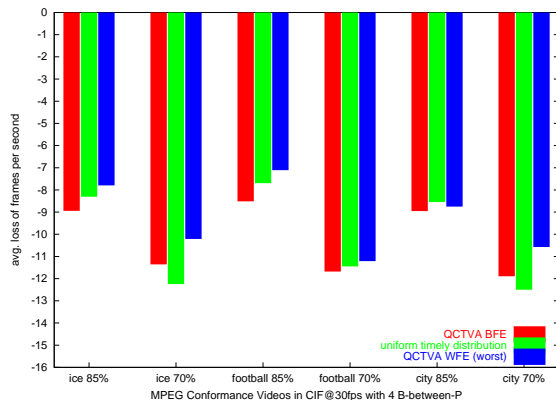


Fig. 3. Frame rate reduction at 85% and 70% of the required average network bandwidth

The reduction to 85% or 70%, respectively, of the required average network bandwidth as underlying Figure 2, is indiscriminately applied to each streamout second (GOP), regardless of the actual size (required bitrate) of the GOP. It is a well-known fact, however, that MPEG-4 is more of a variable bitrate coding scheme than a constant bitrate one. So the average bandwidth does not correctly represent each single video second, and does even less pertain to each single frame. B-frame sizes vary because of varying image differences to their reference frames (e.g., amount of motion between frames) and further by their distance to the reference frames. So whenever different B-frames are chosen for dropping, they offer a varying proportion of bandwidth scalability.

Figure 3 (and a comparison to Figure 2) demonstrates that frame rate reduction is not directly correlated to quality reduction. Note that the frame size of an important B-frame will be proportionally larger, since it has to code more information. Timely uniform frame dropping is more or less random on the sizes of the dropped frames, since it pursues the only goal of uniform dropping over time. So it will discard small-sized but also large-sized frames at will, regardless of their importance. This fact makes it impossible to correlate frame rate with the quality results. QCTVA, with both the BFE and the WFE heuristics, selects priorities only based on their PSNR importance, which ignores frame sizes. QCTVA BFE will

always prefer dropping unimportant and thus smaller frames, so it will therefore have to drop more than QCTVA WFE.

To further optimize visual results, fast prioritization algorithms have to be found which try to optimize quality performance (in terms of PSNR) but also keep in mind the varying frame sizes and therefore can avoid low frame rates.

IV. CONCLUSION AND FUTURE WORK

Our measurements show that QCTVA, based on qualitative PSNR evaluations in the uncompressed video domain, always results in the best quality. Note that this is coupled with extra processing requirements and off-line analysis. If the focus is on fast preparation and decision taking, timely uniform distributed frame dropping always offers at least a good compromise. The more adaptation is needed, the more QCTVA and timely uniform frame dropping converge to the same result, so QCTVA (and the necessary off-line analysis in the uncompressed domain) is only helpful in small adaptation ranges (e.g., 15%). Temporal adaptation at all should be only applied for 15%–30% bandwidth reduction. Beyond that, other means of adaptation should be used. Our future work will try to couple low-range temporal adaptation based on QCTVA with more coarse-grained switching to an alternative video stream as the network becomes severely congested.

REFERENCES

- [1] C. Herpel and A. Eleftheriadis, "MPEG-4 Systems: Elementary Stream Management," *Image Communication Journal. Tutorial Issue on the MPEG-4 Standard*, vol. 15, January 2000.
- [2] D. A. Robinson, "The mechanics of human smooth pursuit eye movement," *The Journal of Physiology*, vol. 180, 1965.
- [3] K. Leopold, H. Hellwagner, and M. Kropfberger, "QCTVA - quality controlled temporal video adaptation," *Proceedings of the SPIE International Symposium ITCOM 2003 on Internet Multimedia Management Systems IV*, vol. 5242, September 2003.
- [4] M. Kropfberger and H. Hellwagner, "Evaluation of RTP Immediate Feedback and Retransmission Extensions," *Proceedings of ICME 2004*, June 2004.
- [5] J. Ott, S. Wenger, N. Sato, C. Burmeister, and J. Rey, "Extended RTP profile for RTCP-based feedback (RTP/AVPF)." draft-ietf-avt-rtp-feedback-04.txt, October 2002. expires April 2003.
- [6] J. Rey, D. Leon, A. Miyazaki, V. Varsa, and R. Hakenberg, "RTP retransmission payload format." draft-ietf-avt-rtp-retransmission-04.txt, December 2002. expires May 2003.