QCTVA - quality controlled temporal video adaptation

Klaus Leopold, Hermann Hellwagner and Michael Kropfberger
Department of Information Technology
University Klagenfurt

{klaus,hellwagn,mike}@itec.uni-klu.ac.at
Universitätsstrasse 65-67, 9020 Klagenfurt, Austria

ABSTRACT

Multimedia streaming is becoming more and more popular. Seamless video streaming in heterogeneous networks like the Internet turns out as almost impossible due to varying network conditions - streams must be adapted to the current network QoS. Temporal scalability is one of the most reasonable adaptation techniques because it is fast and easy to perform. Today's approaches simply drop frames out of a video without spending much effort on finding an intelligent dropping behavior. This usually leads to good adaptation results in terms of bandwidth consumption but also to suboptimal video quality within the given bounds. Our approach offers *analysis of video streams* to achieve the *qualitatively best temporal scalability*. For this reason, we introduce a data structure called *modification lattice* which represents all frame dropping combinations within a sequence of frames. On the basis of the modification lattice, quality estimations on frame sequences can be performed. Moreover, a heuristic for fast and efficient quality computation in a modification lattice is presented. Experimental results illustrate that temporal video adaptation based on QCTVA information leads to a better video quality compared to "usual" frame dropping approaches. Furthermore, QCTVA offers frame priority lists for videos. Based on these priorities, numerous adaptation techniques can increase their overall performance when using QCTVA.

1. INTRODUCTION

Adaptation is becoming more and more important for resource and media management in distributed multimedia systems.[1] Due to bandwidth fluctuations in heterogeneous networks, seamless video streaming turns out to be almost impossible. Thus, video adaptation has to be performed to ensure proper transmission of video data. Temporal scalability turned out to be a promising adaptation technique because it is fast and easy to perform. Common approaches[2, 3] simply drop frames out of a video without spending much effort on finding an intelligent dropping behavior. This usually leads to good adaptation results in terms of bandwidth consumption but also to suboptimal video quality. However, there are certain aspects to take care of when frame dropping is applied:

- Remaining quality after dropping frames
- Timely distribution of frames to be dropped
- Frame sizes
- Importance of dropping candidates

When frame dropping is applied, the video *quality* usually suffers because video sequences do not look smooth. It is the decoder's and video player's job to deal with missing frames. There are sophisticated approaches doing interpolation between the previous and the succeeding frame(s) but usually, video players simply neglect the missing frame and only show the predecessor.[4] Another aspect to take care of when dropping B-Frames is the *timely distribution* of the dropped frames.[5] Usually, the human visual perceptual quality of a frame sequence is much higher if not too many frames are dropped sequentially but in a timely uniform distributed manner. Depending on the video codec, the *frame size variation* might be very high. Dropping one frame might affect the video's bandwidth more than dropping a couple of frames. Usually, the fewer frames are dropped, the smoother the video looks like. Another aspect to take care of by dropping frames out of video sequences might be their *importance*. Several frames carry different experiences like exciting, boring, or X-rated scenes. Applying frame dropping on boring scenes might be better than dropping exciting frames. The meta information governing this decision could be expressed for example by MPEG-7 metadata descriptions.[6]

Liu et al. describes intelligent priorization of frames based on their motion vectors.[7] They pointed out that the human eye and brain is confused by missing frames during supposedly smooth motion. With special motion models they try to spare heavy motion scenes from beeing dropped. Further, they add support for timely distribution, if they have to drop frames even in heavy motion scenes, it still stays smoother than dropping arbitrarily.

This paper proposes analysis of video streams to achieve the *qualitatively best temporal scalability* by measuring the *visual* quality of possible frame dropping combinations. Experimental results illustrate that temporal video adaptation based on QCTVA information leads to a better video quality than random dropping. Furthermore, there are numerous applications and adaptation techniques that can gain profit when using QCTVA.

The remainder of this paper is organized as follows. Section 2 points out basic considerations about temporal scalability. In Section 3 quality measures for frame sequences with respect of temporal scalability are presented. Section 4 presents the main considerations of the QCTVA approach. Section 5 introduces into a technique to determine frame priorities of videos. Experimental results of the QCTVA approach are presented in Section 6. Section 7 is about fields of applications of the QCTVA approach. Ideas about improvements of existing applications are presented. Finally, Section 8 presents conclusions and outlines future work.

2. BASIC CONSIDERATIONS ABOUT TEMPORAL SCALABILITY

In MPEG-4 each video elementary stream[8] is defined as a sequence of VOPs (*video object planes*), i.e., frames. Three important video frame types are distinguished: I-VOPs, P-VOPs, and B-VOPs. I-frames are independent from any other frames, P-frames are based on predictions from the last reference frame, and B-frames are based on predictions from the previous and the following reference frames. A reference frame might be either an I-frame or a P-frame, so only B-frames are totally unreferenced by any other frame type. When performing temporal video adaptation, the frame rate of a video is reduced. Thus, temporal scaling can be seen as a variation of the video in the time domain. One can perform temporal video adaptation by separating the video stream into two layers: base and enhancement layer.[9] The base layer carries a subset of the video frames of the video. The enhancement layer adds frames and therefore increases the frame rate. Clients can now choose whether they would like to receive the video with the low or high frame rate. However, layered temporal video adaptation is rather coarse grain because the adaptation granularity is a whole layer consisting of many frames. Furthermore, temporal video adaptation can be performed in the *compressed* or in the *uncompressed* domain.[10] Since there are no decoding dependencies, frame dropping in the uncompressed domain is simple because any frame can be dropped. In the compressed domain, decoding dependencies exist and thus, not every frame can be dropped. The QCTVA approach performs frame dropping in the compressed domain. The influence on the video by dropping a certain frame type can be summarized as follows:

- B-frames can be dropped at will since there are no other frames referencing them.
- When dropping a P-frame F_p all previous B-frames forward-referencing F_p and also all following B-frames and P-frames backward-referencing F_p have to be dropped.
- Dropping I-frames means losing all following P- and B-frames until the next I-frame as well as losing all forward-referencing P- and B-frames. Since I-frames are usually infrequently used in a frame pattern, dropping I-frames makes nearly no sense.

Due to the independence of B-frames it is quite easy to randomly drop them.[2]

3. QUALITY ESTIMATION FOR GOPS

To evaluate the quality of a frame pattern, the average Peak Signal-to-Noise Ratio (PSNR)[11] is calculated for every frame. A pattern represents a frame sequence of arbitrary length which is constant over the whole video. In this context, GOP (group of pictures) and pattern are synonyms and therefore used interchangeably. Signal-to-noise ratio measures are estimates of the quality of a reconstructed image compared with an original image. The basic idea is to compute a single, objective number that reflects the quality of the reconstructed image. The QCTVA approach uses signal-to-noise ratio measures because they are fast and easy to compute.

Quality of GOPs. The PSNR value for two images (frames) is computed by

$$psnr(I, L) = 20 \log_{10} \left(\frac{255}{\sqrt{\frac{1}{XY} \sum_{x=1}^{X} \sum_{y=1}^{Y} \left(I_{x,y} - L_{x,y}\right)^2}} \right) \tag{1}$$

where I is the original and L is the loss induced image with the dimension $X \times Y$ each and 8-bit pixels. To get the loss induced image, the original image is encoded and afterwards decoded.

Given two patterns F (the original frame sequence) and G (the loss induced frame sequence) with n single frames $F = \{F_1, \dots, F_n\}$ and $G = \{G_1, \dots, G_n\}$, the quality Q_P of G relative to F is the average PSNR value of the pattern which is computed by:

$$Q_P = \frac{\sum_{i=1}^{n} psnr(F_i, G_i)}{n} \tag{2}$$

Quality of GOPs with Dropped Frames. The above mentioned quality estimation can be performed on patterns if no frames are missing. Dropping I- and P-frames usually makes no sense because the loss in quality is rather large due to the dependencies to other frames. Therefore, equations only for dropping B-Frames will be developed next. For simplicity, a dropped B-frame is also referred to just as a frame.

The quality of patterns with a single dropped frame is calculated by using the last available frame instead of the dropped frame. Given two patterns F and G with n frames, where frame k is missing in pattern G, the quality Q_P is calculated by:

$$Q_P = \frac{\sum_{i=1}^{k-1} psnr(F_i, G_i) + psnr(F_k, G_{k-1}) + \sum_{j=k+1}^{n} psnr(F_j, G_j)}{n} \tag{3}$$

It is not possible to handle patterns where more than one frame is dropped with Equation 3. With Equation 4, the quality of a pattern with a sequence of dropped frames can be calculated. Given two patterns F and G with n frames where m frames are sequentially dropped starting with frame G_k, the quality Q_P is calculated as

$$\begin{aligned} &\forall\, k, m, n \in \mathbb{N} : k + m \leq n + 1 \\ &Q_P = \frac{\sum_{i=1}^{k-1} psnr(F_i, G_i) + \sum_{l=k}^{k+m-1} psnr(F_l, G_{k-1}) + \sum_{j=k+m}^{n} psnr(F_j, G_j)}{n} \end{aligned} \tag{4}$$

If more than one sequence of frames is dropped in a pattern (e.g., frames $3, 4, 5, 7$, and 10 are dropped), Equation 4 can not be applied. For this case Equation 5 is defined. F and G are patterns with length n, k is the starting index of the first frame dropping sequence with m frames, k' is the starting index of the succeeding frame dropping sequence with m' frames etc., $k^{(n)}$ is the starting index of the last frame dropping sequence with $m^{(n)}$ frames.

$$\begin{aligned} \forall\, k, k', \dots, k^{(n)}, m, m', \dots, m^{(n)}, n \in \mathbb{N} :\ & 1 < k < k + m < k' < k' + m' < k^{(n)} \\ & < k^{(n)} + m^{(n)} \leq n + 1 \end{aligned}$$

$$\begin{aligned} & Q_P \left(F, G, \left[(k, m), (k', m'), \dots, \left(k^{(n)}, m^{(n)}\right)\right], n\right) \\ & = \frac{\sum_{i=1}^{k-1} psnr\left(F_i, G_i\right) + \sum_{l=k}^{k+m-1} psnr\left(F_l, G_{k-1}\right)}{n} \\ & + \frac{\sum_{i=k+m}^{k'-1} psnr\left(F_i, G_i\right) + \sum_{l=k'}^{k'+m'-1} psnr\left(F_l, G_{k'-1}\right)}{n} \\ & + \dots \\ & + \frac{\sum_{i=k^{(n)-1}+m^{(n)-1}}^{k^{(n)}-1} psnr\left(F_i, G_i\right) + \sum_{l=k^{(n)}}^{k^{(n)}+m^{(n)}-1} psnr\left(F_l, G_{k^{(n)}-1}\right)}{n} \\ & + \frac{\sum_{i=k^{(n)}+m^{(n)}}^{n} psnr\left(F_i, G_i\right)}{n} \end{aligned} \tag{5}$$

The start points and lengths of dropped frame sequences are specified as a list of tuples: $[(k, m), (k', m'), (k'', m''), \ldots]$. Equation 5 basically computes the average PSNR value of a frame sequence without dropped frames followed by a sequence including dropped frames and so on. Thus, all $k, k', \ldots$ and $m, m', \ldots$ must hold the condition $k < k + m < k' < k' + m' \ldots$. The last term computes possible remaining frame sequences without dropped frames. If there are no remaining frame sequences, $k^{(n)} + m^{(n)} > n$ and thus, the term becomes 0.

Table 1 presents an example PSNR estimation. Column one represents the frame number and column two the frame type of the original pattern. Column three and four represent the frame number and type of the sequence with frames dropped. Column five shows the computed PSNR value for each frame pair. Shaded rows represent dropped frames in G.

No_F	$Type_F$	No_G	$Type_G$	$psnr(F, G)$
1	I	1	I	35.342
2	B	2	B	33.993
3	B	2	B	32.984
4	B	2	B	31.191
5	B	2	B	29.032
6	P	6	P	35.561
7	B	6	P	32.432
8	B	8	B	34.331
9	B	9	B	34.531
10	B	9	B	31.667
11	B	11	P	34.123
Overall Q_P				33.198

Table 1. PSNR Estimation in Case of Randomly Dropped B-Frames

The arguments for Equation 5 can be derived from Table 1: k and m are both 3 because the starting index for the first dropping sequence as well as its length is 3, $k' = 7$ because this is the start of the next frame dropping sequence with length $m' = 1$, $k'' = 10$, $m'' = 1$, and $n = 11$ because the whole pattern contains 11 frames. Equation 6 is an example for applying Equation 5 on Table 1.

$$\begin{aligned} &Q_P(F, G, [(3,3), (7,1), (10,1)], 11) \\ &\quad = \frac{\sum_{i=1}^{2} psnr(F_i, G_i) + \sum_{l=3}^{5} psnr(F_l, G_2)}{11} \\ &\quad + \frac{\sum_{i=6}^{6} psnr(F_i, G_i) + \sum_{l=7}^{7} psnr(F_l, G_6)}{11} \\ &\quad + \frac{\sum_{i=8}^{9} psnr(F_i, G_i) + \sum_{l=10}^{10} psnr(F_l, G_9)}{11} \\ &\quad + \frac{\sum_{i=11}^{11} psnr(F_i, G_i)}{11} = 33.198 \end{aligned} \tag{6}$$

4. THE QCTVA APPROACH

Modification Lattice. A single frame sequence has a lot of different dropping patterns. For example, the pattern `IBBPBB` may have the dropping sequences `I-BPBB`, `I--PBB`, or even `I--P--`, where `-` represents a dropped frame. To compute the quality of any frame dropping sequence, a *modification lattice* of all frame dropping combinations has to be built. The original sequence is the *master pattern* and its frame dropping sequences are referred to as *modifications*. All modifications with the same number of dropped frames are labeled as a *layer*. The modification where no more B-frames are available is called the *base layer*. If n is the number of droppable frames (B-frames), the modification lattice has n layers and its height is n. We do not count the master pattern as a layer because it is the origin of the lattice - the original frame sequence of the video. Layer i in the lattice represents all combinations of i frames being dropped from the master pattern.

Figure 1 illustrates a modification lattice with the master pattern `IBBPBB`. Layer 1 in the lattice represents all possible frame dropping combinations of the master pattern by dropping only one frame for each modification. This leads to the modifications for layer 1 (L_1): `I-BPBB`, `IB-PBB`, `IBBP-B` and `IBBPB-`. For layer 2 (L_2) all modifications of L_1 are expanded by dropping one more frame. Thus, e.g., the pattern `IB-PBB` will lead to `I--PBB`, `IB-P-B` and `IB-PB-` at layer 2.

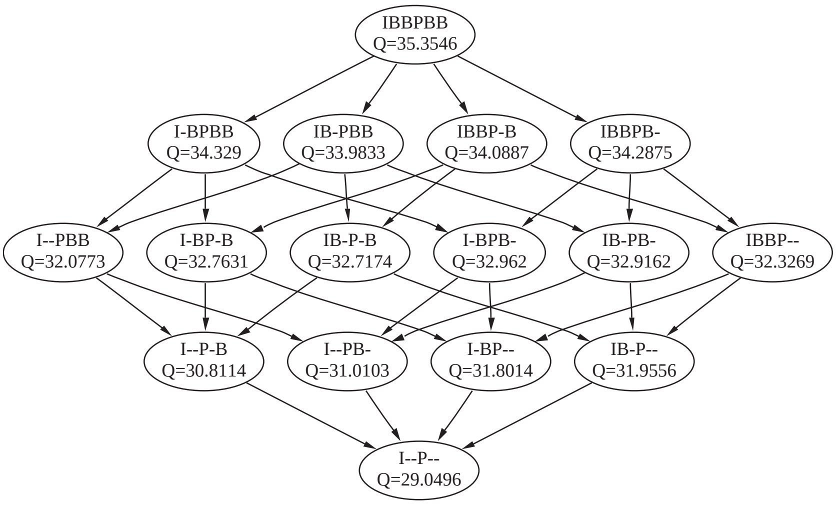


Figure 1. Modification Lattice Including Quality Measures

A quality value Q is assigned to every node (modification) in the lattice. For the master pattern Q is computed based on Equation 2, where F is the original (lossless) frame sequence and G is the loss induced master pattern. For the other nodes Q is estimated by applying Equation 5 where F is the original (lossless) frame sequence and G is the loss induced modification (node) in the lattice. The lattice in Figure 1 is referred to as fully expanded because all nodes in the lattice are expanded. A fully expanded lattice represents *all* possible frame dropping combinations for a given master pattern.

Best and Worst Modifications Estimation. To extract dropping information from the lattice one has to perform certain operations on the lattice. Interesting information might be the best and worst patterns on each layer which leads to the best and worst dropping behavior respectively. The worst modifications might be used for reasons of comparison only.

The set of best patterns in a modification lattice contains the master pattern, the patterns with the *highest* quality measure on each layer, and the base layer. Given a lattice with n layers $L_1 \ldots L_n$, the master pattern MP, and the base layer $BL = L_n$, the set of best patterns P_B is defined as:

$$\begin{aligned} \forall\, i, n \in \mathbb{N} \quad & : i = 1 \ldots (n-1) \\ P_B \quad & = \{MP\} \cup maxQ(L_i) \\ maxQ(L_i) \quad & = max(L_i) \cup maxQ(L_{i+1}) \\ maxQ(L_n) \quad & = \{BL\} \end{aligned} \tag{7}$$

The maximum operation $max(L_i)$ of Equation 7 takes a set of patterns (nodes in the lattice) as input and returns the *set of patterns* with the maximum quality. Figure 2 graphically illustrates the results of applying Equation 7 on the lattice in Figure 1. The best patterns in the lattice are emphasized. The worst modifications are estimated the same way as the best modifications except that the functions max and $maxQ$ of Equation 7 are replaced by min and $minQ$.

Average Modification Estimation. Average modifications of a modification lattice simulate the long time behavior of semi-intelligent network nodes. A semi-intelligent network node would not drop I- and P-frames if bandwidth back-offs occurred, but choose B-frames only. Given a layer L in a lattice with m modifications, the average quality measure is computed by:

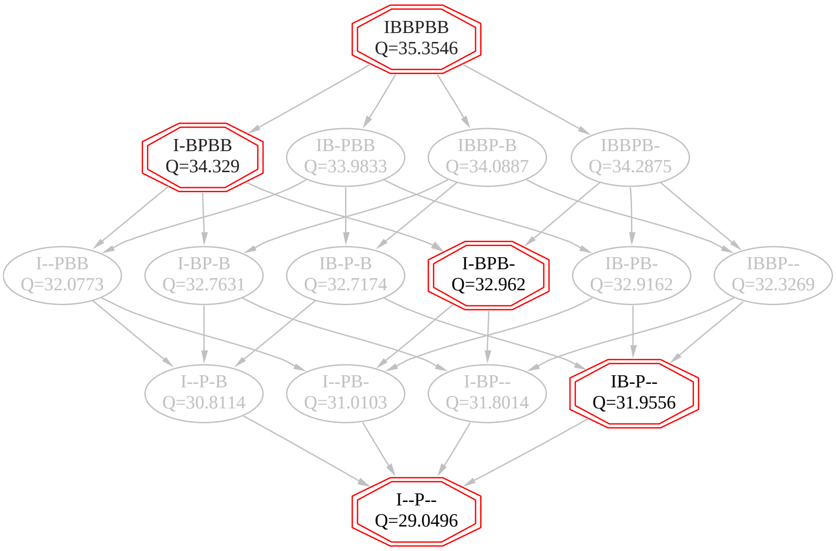


Figure 2. Best Patterns in a Lattice

$$Q_{avg}(L) = \frac{\sum_{i=1}^{m} Q_{P_i}}{m} \tag{8}$$

The average modification on a layer is the pattern with the minimum deviation of the average quality measure at this layer. Given a layer L in a lattice with m modifications, the average modification P_{A_i} is computed by:

$$avg(L) = \{P_{A_i} \mid i \in \{1, \ldots, m\} : min\left(|Q_{avg}(L) - Q_{P_1}|, \ldots, |Q_{avg}(L) - Q_{P_m}|\right)\} \tag{9}$$

The set of average patterns in a lattice contains the master pattern, the patterns with the minimum deviation to the *average* quality measure on each layer, and the base layer. Given a lattice with n layers $L_1 \ldots L_n$, the master pattern MP, and the base layer $BL = L_n$, the set of average patterns P_A is defined as:

$$\begin{array}{rl} \forall\, i, n \in \mathbb{N} & : i = 1 \ldots (n-1) \\ P_A & = \{MP\} \cup avgQ(L_i) \\ avgQ(L_i) & = avg(L_i) \cup avgQ(L_{i+1}) \\ avgQ(L_n) & = \{BL\} \end{array} \tag{10}$$

Figure 3 graphically illustrates the result of applying Equation 10 on the lattice in Figure 1. The average patterns in the lattice are colored red.

Best First Expansion Heuristic. The Best First Expansion Heuristic (BFE) defines a very fast way to build a modification lattice and offers a continuous path through the lattice. A continuous path provides a total order of all modifications. The principle is not to fully expand the lattice but the best nodes on each layer only. The starting point is the master pattern which is expanded to all its child patterns. Then, quality estimation is performed. In the set of expanded nodes only the qualitatively best pattern is expanded further - the others are not. This principle is repeated for all layers until the base layer is reached. This method is referred to as *Best First Expansion Heuristic* (BFE).

Given a lattice where MP is the master pattern, BL is the base layer pattern and N is a singleton set containing any node in the lattice, then the set of best first expansion nodes P_H is defined as

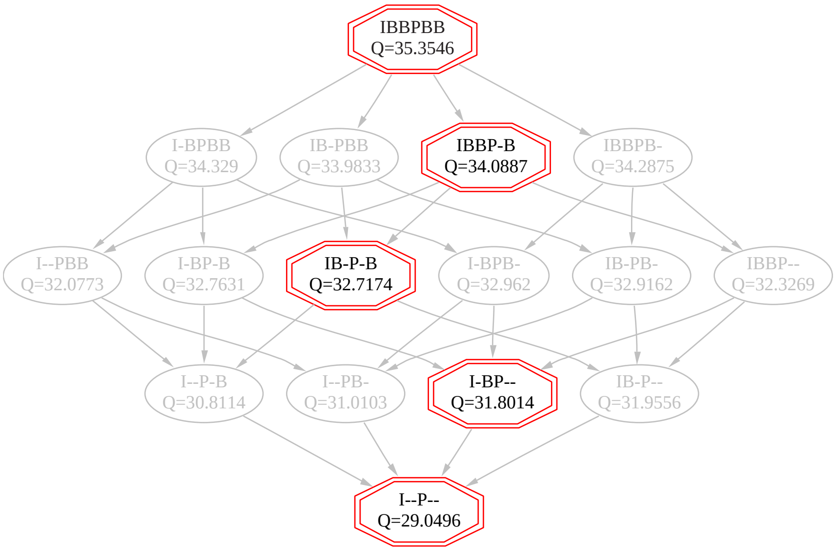
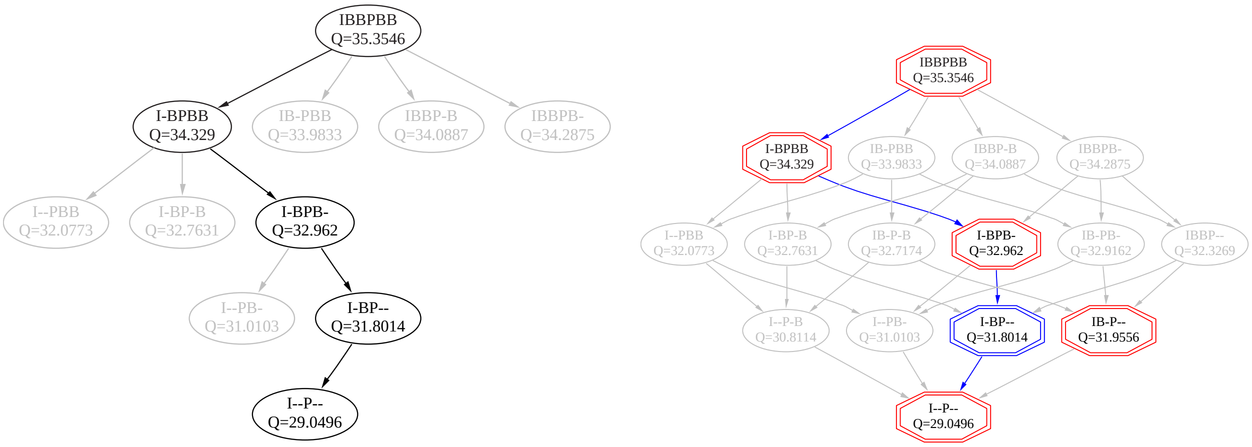


Figure 3. Average Patterns in a Lattice



(a) BFE Construction

(b) BFE Suboptimal Result

Figure 4. Best First Expansion Heuristic

$$
\begin{aligned}
P_H &= P'_H(\{MP\}) \\
P'_H(N) &= \begin{cases} N \cup P'_H(max(expand(N))) & N \neq BL, \\ N & N = BL \end{cases}
\end{aligned}
\tag{11}
$$

The function $expand(N)$ in Equation 11 expands the node in the singleton set N and returns a set of all its children. The maximum operation $max(M)$ takes a set of patterns M as input and returns the *singleton set of patterns* with the maximum quality. If more than one maximum node exists, one pattern is randomly chosen as the maximum. Figure 4(a) illustrates the result of Equation 11 applied to the lattice in Figure 1.

Best first expansion leads to a path through the lattice. A path is characterized by the constraint that every modification M_i is a predecessor of all derived modifications $M_j \dots M_n$, where i is the current layer, n is the number of layers, and

```
No  Type    Prio  PSNR      Size  FrOffset
------------------------------------------
 0   I-VOP     1   29.0496   7908        0
 1   P-VOP     2   29.0496   2677     7908
 2   B-VOP     6   35.3546   1579    10585
 3   B-VOP     3   31.1014   1540    12164
 4   P-VOP     2   29.0106   2785    13704
 5   B-VOP     4   32.962    1538    16489
 6   B-VOP     5   34.329    1485    18027
 7   P-VOP     2   29.0106   2810    19512
                               . . .
```

Table 2. Frame Prioritization Output

$i < j \leq n$. A modification M_i is a predecessor of a modification M_{i+1} if every dropped frame in M_i is also dropped in M_{i+1}. For example the modification `I-BPBB` is a predecessor of `I-BPB-` but not of `IBBP--`. A path through the lattice also implies a total order of modifications which is needed for frame prioritization as described in Section 5.

Best first expansion may lead to a suboptimal path and to suboptimal modifications. It is possible that the best modification on layer L_i is expanded but its children on layer L_{i+1} do not contain the best modification of layer L_{i+1}. Figure 4(b) illustrates the problem of best first expansion. On layer L_2 pattern `I-BPB-` is expanded which leads to its children `I--PB-` and `I-BP--`. The pattern `I-BP--` is taken for the next expansion because it has the higher quality measure. If the whole lattice would have been expanded, the pattern `IB-P--` would also be in the set of patterns on layer L_3 and it would be chosen as the best modification because it reflects the maximum quality value on this layer.

To find the very best path some sort of best path search algorithm like Dijkstra's must be used.[12] Measurements showed that the modifications determined by the best first expansion heuristic are usually congruent to the best modifications in the lattice. Just very few deviations are being discovered (see Section 6).

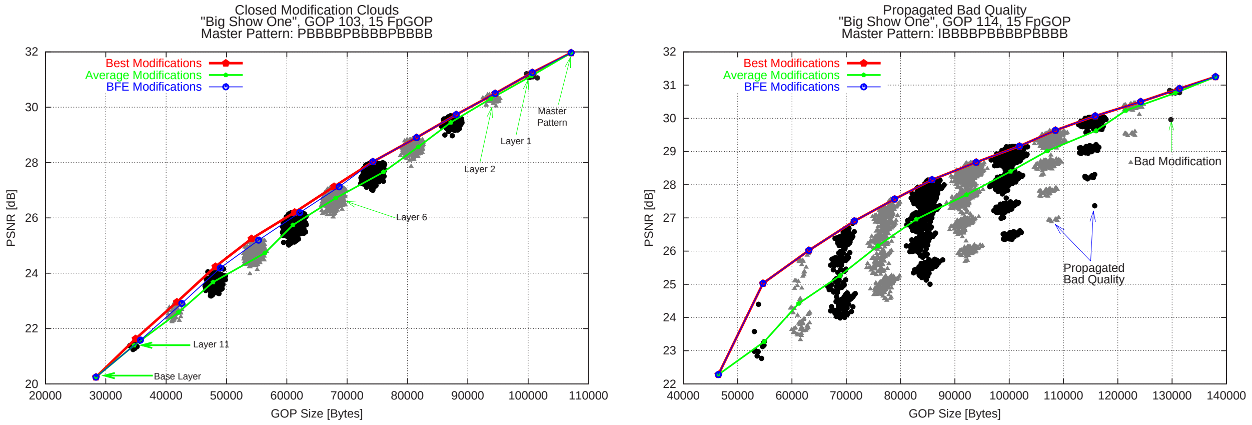
5. FRAME PRIORITIZATION

Based on the best path in the lattice, priorities for 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 layer L_n, priority 4 to the B-frame which is dropped at layer L_{n-1} and so forth. To determine priorities for the B-frames in a pattern, a continuous path through the lattice is needed. A path implies that every modification on a certain layer is a predecessor of all further modifications. Given the modification `IB-PBB` on layer L_1 and the modification `I-BP-B` on layer L_2, it is not possible to assign a priority for the second B-frame because it does not appear on layer L_1 but on layer L_2. In Figure 4(a), the priority for the frames in the base layer is 1 for the I-frame and 2 for the P-frames. 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-` the fifth frame is new and therefore it gets the priority 4. Analogous, the last B-frame in `I-BPBB` gets 5 and the second frame in `IBBPBB` gets 6 as its priority.

The QCTVA tool produces the output shown in Table 2. The first column in the table represents the frame number in stream order because that is the way how frames are transmitted over networks.[9] The second column represents the frame type which can be I-VOP, P-VOP, or B-VOP. The third column represents the priority of the frames based on the considerations above. The PSNR value is the quality measure that can be reached if the current frame and all its lower priorities are decoded. The fifth column represents the size of the frame in bytes and the sixth column the byte offset in the video file. It has to be mentioned that Table 2 shows a human readable form of the frame priorities. The binary representation can be coded in just one byte per frame because all the information like `No`, `Type`, `Size`, and `FrOffset` can be omitted for video delivery.

6. EXPERIMENTAL RESULTS

Extensive experiments were performed to analyze the best frame dropping behavior within video sequences.[13] This section presents results of the GOP analysis where every single GOP of a video was investigated. Furthermore, the streaming



(a) Closed Modification Clouds

(b) Propagated Bad Quality

Figure 5. GOP Analysis

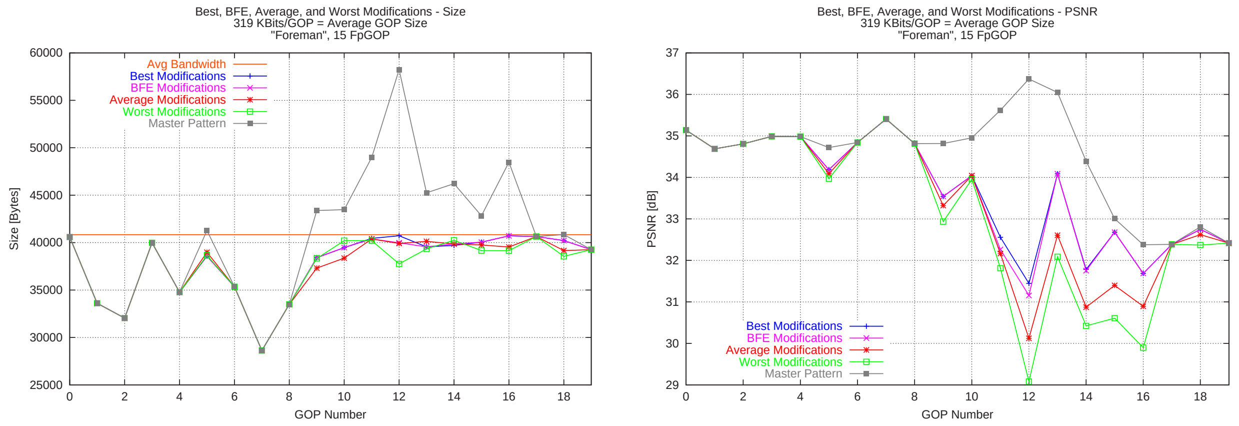
behavior of temporally adaptated videos was analyzed. The MPEG reference videos “Foreman” and “Big Show One” were used as test streams.

GOP Analysis. Figures 5(a) and 5(b) visualize all modifications in a GOP. The x-axis represents the GOP size in bytes and the y-axis the PSNR value in dB. Every black or grey dot stands for a certain modification (i.e., a node in the modification lattice) of the GOP. Black and grey is used alternately to distinguish modifications among layers, i.e., black dots represent layer L_i, grey dots layer L_{i+1}, black L_{i+2}, etc. The most upper points represent the best modification for every layer, the lowest points the average modification, and the points in in between or overlaying the upper points are the modifications based on the BFE heuristic. All points are connected to give a better visualization - *there are no measuring points interpolated in between*. Figure 5(a) shows compact and small modification ”clouds”. The difference between the modification with the highest byte size and the lowest size of a layer is approximately 1000 bytes (e.g., layer 1 or 11) to 4000 bytes (e.g., layer 6). The fluctuation of the PSNR value is about 1 dB per layer. The benefit of quality based temporal adaptation is not very high in comparison to non quality based adaptation. Furthermore, Figure 5(a) shows that the BFE heuristic is not always as good as the best modification. However, the difference in the PSNR value is negligible.

Figure 5(b) represents a GOP consisting of 15 frames where a big benefit can be gained by using quality controlled temporal adaptation. The PSNR value fluctuates up to 3.5 dB which means choosing a wrong pattern may lead to big losses in visual quality. Each vertically aligned bar of modification clouds in the graph represent a single layer, and hereby all layers are visually well separated from each other. The graph also shows modifications which are somewhat clustered within a layer, e.g., layer 3 shows four broad separated modification clouds. The first ranges from approx. 29.5 dB to 30.1 dB, the second from 28.9 to 29.2, etc. This *intra layer modification clustering* is caused by dropping important frames which also leads to a propagation of bad quality to other layers. Assume the master pattern is `IBBBBPBBBB`, the modification `I-BBBPBBBB` carries the best quality and `IBBBBPBBB-` has the worst quality. Succeeding modifications of `I-BBBPBBBB` will be in the upper clouds whereas children of `IBBBBPBBB-` will be in the lower clouds. Furthermore, one can see that the best modifications are congruent with the modifications based on the BFE heuristic.

Streaming Behavior. Figure 6(a) illustrates the bandwidth adaptation of the best, BFE, worst, and average modifications to the average bandwidth. The x-axis represents the GOP number and on the y-axis the size of the GOPs in bytes is assigned. The highest line illustrates the bandwidth consumption of the whole video (i.e., the master pattern). Assume the different modifications are streamed with the same *constant* average bit rate of 319 KBits/GOP. One can see that the consuming bandwidth highly exceeds the average bandwidth starting with GOP 9. Thus, adaptation has to take place. All modifications - best, BFE, average, and worst - do adapt very good to the average bandwidth with quite small deviations.

Figure 6(b) illustrates the adaptation in terms of quality. The video is again streamed with the average GOP size of 319 KBits/GOP. The highest line represents the master pattern’s quality over the streaming period. When comparing with



(a) Bandwidth Adaptation

(b) Quality when Streaming Average GOP Size

Figure 6. Streaming Behavior

Figure 6(a) one can see that the quality decreases every time the video's bandwidth exceeds the average bandwidth. The figure shows that the best modifications are really better than the average and worst modifications. Furthermore, one can see that BFE modifications are always as good as best modifications except in two cases. But even there the deviation in visual quality is diminishingly small so that it can not be observed by the human eye.

7. FIELDS OF APPLICATIONS

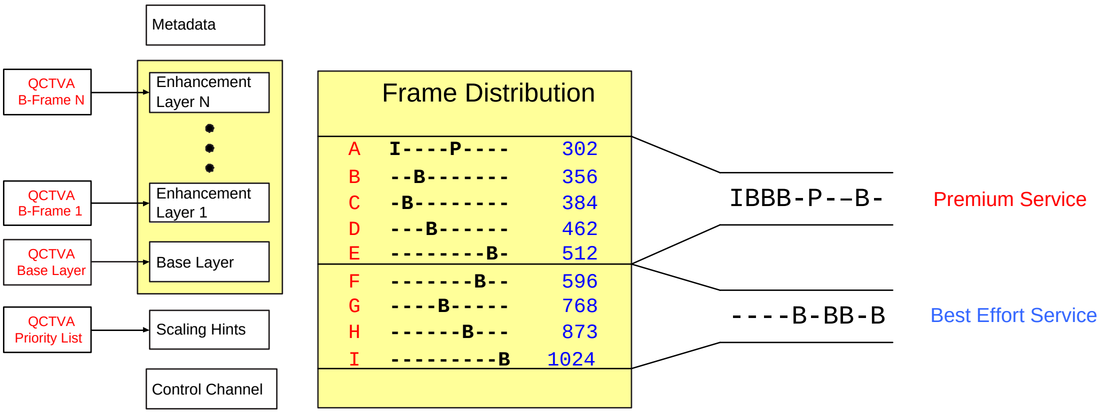
There are numerous adaptation techniques and applications that can gain of using QCTVA. This section outlines areas where QCTVA can be integrated, exemplified by the video adaptation proxy/gateway QBIX,[14] the adaptation-aware multimedia streaming protocol AMSP,[15] and the network technology Differentiated Services (DiffServ).[16] Other fields of applications such as video servers, receiver driven layered multicast (RLM),[17] and video stream switching are conceivable.[13]

Video Adaptation Proxies. Video adaptation proxies are used to improve cache replacement strategies by quality aware caching.[18] This means that a video is not fully replaced from the cache but its quality is reduced. Furthermore, proxies can fulfill the functionality of media gateways which means that on-the-fly video adaptation or transcoding is performed. QBIX[14] fulfills the functionality of a multimedia proxy and multimedia gateway. The QBIX approach offers the concept of adaptation chains which is a mechanism where an incoming video passes through different adaptation steps. Currently, QBIX offers temporal and spatial adaptation as well as grey scaling. The temporal adaptor drops *all* B-frames regardless of their quality. Removing all B-frames in terms of quality aware caching might be a reasonable strategy but not if the proxy has to fulfill the functionality of a media gateway. Dropping all B-frames is a rather coarse grain adaptation which leads to high quality and bandwidth variations.

The temporal adaptor of the QBIX project could be improved by using QCTVA frame prioritization information. If the gateway has to perform temporal adaptation, it makes a prioritization table look-up and drops the least important frames first with respect to the client's bandwidth.

AMSP - The Adaptation-Aware Multimedia Streaming Protocol. The adaptation-aware multimedia streaming protocol (AMSP)[15] is a network protocol by supporting applications with the transport of multimedia data. It provides *channels* with different priorities to separate layered streams. The multimedia content can be mapped to one ore more media channels (MCs). If a bandwidth back-off occurs during the transmission, AMSP starts to drop MCs starting with the one with the lowest priority.

The AMSP architecture is very "QCTVA friendly" which means that the frame prioritization scheme of QCTVA can be directly mapped onto AMSP channels. Figure 7(a) illustrates the mapping. AMSP features one base layer channel and



(a) AMSP with QCTVA

(b) DiffServ with QCTVA

Figure 7. Fields of Applications

multiple enhancement layer channels. The AMSP base layer could be filled with the frames in the QCTVA base layer. The first AMSP enhancement layer is equivalent to the highest priority B-frame of QCTVA and so forth. Thus, enhancement layer 1 up to enhancement layer N are being filled with B-frames starting with the highest priority frame. Furthermore, AMSP's $Scaling\ Hints$ could be filled with the QCTVA priority list (see Table 2) which implies a propagation of QCTVA information to multiple nodes in the network.

Differentiated Services. Another application could be Differentiated Services (DiffServ).[16] DiffServ offers the ability to classify the traffic based on service level agreements. Different service classes with soft bandwidth guarantees are available. The "Premium Service" for example offers a fixed maximum bandwidth which is available when needed. One could assign frames based on QCTVA prioritization lists to the premium class until its bandwidth is exceeded. The remaining frames could be transferred on a best effort channel. Figure 7(b) illustrates the frame assignment to DiffServ service classes. The box represents the assignment of frames to priority classes. Class A has the highest priority and class I the lowest. Assume that a DiffServ premium service with 512 KBits/sec and a best effort service are available. The frames in the classes A to E are assigned to the premium service because they are consuming not more than 512 KBits/sec. Frames in the classes F to I are assigned to the best effort service.

8. CONCLUSION AND FUTURE WORK

This work showed that temporal video scalability has to be more than just randomly dropping frames. Spending effort in finding a good frame dropping behavior significantly enhances video quality if temporal adaptation has to be performed. Quality estimation methods based on the well known peak signal-to-noise ratio (PSNR) value for GOPs with missing frames were presented. The computed qualities are the most important measured values for the modification lattice - a graph with all possible frame dropping combinations for a GOP. The best first expansion heuristic (BFE) provides a fast way of finding a path through the lattice which allows the assignment of priorities to single frames. Using frame priority lists of the QCTVA approach improves different video streaming adaptation techniques and can be integrated in certain applications to enhance their functionality.

Future work will be the investigation of the interplay of QCTVA and AMSP.[15] The expected benefit of the combination of these two approaches is a fully temporal adaptive video streaming solution supporting a wide range of bandwidths. The QCTVA frame priority lists will also be integrated in the QBIX multimedia proxy/gateway,[14] replacing the existing coarse grain temporal video adaptor with an intelligent frame dropping mechanism. A compact binary representation of the priority lists has to be developed for streaming the lists together with the video data. Expected benefits are fine grained and high-quality adaptation and cache replacement strategies.

Further, the approach of judging single frames could be extended by Fine Grained Scalability (FGS) coding,[19] which adds an enhancement layer for additional quality to every frame. Still, the frames themselves could be either I-VOPs, P-VOPs or B-VOPs. This could allow us to pre-calculate even more detailed priority levels like ”drop frame n completely but keep frame $n+1$ with 50% of its size“. Hereby we also realize more fine grained scalability steps, which easily would allow us to adapt the ideas of multicast-capable layers like discussed in Rejaie et al.[20]

Special attention for future work has to be given to real-time (and even faster) QCTVA calculation, introduced latency and/or simple and efficient transmission of pre-calculated priorities to proxies and gateways. Frame priorities could be expressed by standardized means of MPEG-21 digital items[21] .

REFERENCES

1. L. Böszörmenyi, H. Hellwagner, H. Kosch, M. Libsie, and S. Podlipnig, “Metadata Driven Adaptation in the ADMITS Project,” *EURASIP Signal Processing: Image Communication, Special Issue on Multimedia Adaptation* , 2003 (to appear).
2. S. Cen, C. Pu, R. Staehli, C. Cowan, and J. Walpole, “A Distributed Real-Time MPEG Video Audio Player,” in *Fifth International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV’95)*, pp. 142–153, April 1995.
3. L. Rowe, K. Patel, B.C.Smith, and K.Liu, “MPEG Video in Software: Representation, Transmission and Playback,” in *Proc. of SPIE - International Society of Optical Engineering*, pp. 134–144, February 1994.
4. D. Scharsten, “Synthesizing New Views from Stereo Data,” *IEEE Workshop on Representations of Visual Scenes (in conjunction with ICCV’95)* , 1995.
5. M. Hemy, U. Hengartner, P. Steenkiste, and T. Gross, “MPEG System Streams in Best-Effort Networks,” in *Packet Video Workshop 99*, May 1999.
6. J. M. Martinez, “Overview of the MPEG-7 Standard,” *ISO/IEC JTC1/SC29/WG11 N4031* , March 2001.
7. T. Liu, H. Zhang, and F. Qi, “Perceptual frame dropping in adaptive video streaming,” *IEEE International Symposium on Circuits and Systems (ISCAS)* , May 2002.
8. C. Herpel and A. Eleftheriadis, “MPEG-4 Systems: Elementary Stream Management,” *Image Communication Journal. Tutorial Issue on the MPEG-4 Standard* **15**, January 2000. http://leonardo.telecomitalialab.com/icjfiles/mpeg-4_si/.
9. F. Pereira and T. Ebrahimi, eds., *The MPEG-4 Book*, Prentice Hall PTR, 2002.
10. M. Libsie and H. Kosch, “Content Adaptation for Multimedia Indexing Retrieval,” Tech. Rep. TR/ITEC/02/2.08, University Klagenfurt, June 2002.
11. J. L. Mannos and D. J. Sakrison, “The Effects of a Visual Fidelity Criterion on the Encoding of Images,” *IEEE Trans. Information Theory* **20(4)**, pp. 525–536, 1974.
12. R. Sedgewick, *Algorithms in Modula-3*, Addison-Wesley Publishing Company Inc., 1993.
13. K. Leopold, “Quality Controlled Temporal Video Adaptation,” Master’s thesis, University Klagenfurt, January 2003.
14. P. Schojer, L. Böszörmenyi, H. Hellwagner, B. Penz, and S. Podlipnig, “Architecture of a Quality Based Intelligent Proxy (QBIX) for MPEG-4 Videos,” *WWW2003* , May 2003.
15. M. Ohlenroth and H. Hellwagner, “A Protocol for Adaptation-Aware Multimedia Streaming,” *International Conference on Multimedia and Expo (ICME)* , July 2003.
16. X. Xiao and L. Ni, “Internet QoS: A Big Picture,” *IEEE Network* **13**, pp. 8–18, March/April 1999.
17. S. McCanne, V. Jacobson, and M. Vetterli, “Receiver-driven Layered Multicast,” in *ACM SIGCOMM*, pp. 117–130, ACM Press, August 1996.
18. S. Podlipnig and L. Boeszoermenyi, “Replacement Strategies for Quality Based Video Caching,” in *International Conference on Multimedia and Expo (ICME)*, **Volume 2**, pp. 49–53, (Lausanne, Switzerland), August 2002.
19. W. Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard,” *IEEE Trans. Circuits and Systems for Video Technology* **11**, March 2001.
20. R. Rejaie, M. Handley, and D. Estrin, “Layered Quality Adaptation for Internet Video Streaming,” *IEEE JSAC. Special Issue on Internet QoS.* , Winter 2000.
21. J. Bormans and K. Hill, eds., *MPEG-21 Overview, ISO/IEC JTC1/SC29/WG11 N5231*, October 2002. http://mpeg.telecomitalialab.com/standards/mpeg-21/mpeg-21.htm.