Towards User-driven Adaptation of H.264/SVC Streams

Jordi Ortiz Murillo
jordi.ortiz@um.es

Michael Ransburg
mransbur@itec.uni-klu.ac.at

Eduardo Martínez Graciá
edumart@um.es

Michael Sablatschan
msablats@itec.uni-klu.ac.at

Antonio F. Gómez Skarmeta
skarmeta@um.es

Hermann Hellwagner
hellwagn@itec.uni-klu.ac.at

Keywords

H.264/SVC, Adaptation, TFRC, QoE, User-driven, Streaming

ABSTRACT

H.264/SVC enables runtime-efficient scalability in the spatial, temporal and fidelity dimension. Existing adaptation mechanisms facilitate this to automatically adapt the H.264/ SVC stream to the current usage environment without any user interaction. This paper argues that the Quality of Experience (QoE) of the end user can be enhanced by enabling him to manually adjust the adaptation if he wishes to do so. An approach which enables this is presented and evaluated. It is shown that by facilitating this approach an increased QoE is provided compared to automatic adaptation approaches. Finally, future work indicates the next steps in order to implement this approach.

1. INTRODUCTION

H.264/SVC [1][2] is a scalable video codec, which introduces scalability mechanisms in three different scalability dimensions: spatial resolution, temporal resolution and fidelity. The scalability of the encoded video bit-stream is achieved by a layered approach. A scalable H.264/SVC bit-stream comprises an H.264/AVC-conformant base layer, which represents video at the lowest quality, and one or more enhancement layers, which can be used to refine the video quality in one or more of the above mentioned scalability dimensions. The adaptation of an H.264/SVC bit-stream is performed by simply truncating these enhancement layers from the initial bit stream and can therefore be implemented very efficiently anywhere along the delivery chain. H.264/SVC thus fits the requirements of video streaming in heterogeneous usage environments and is therefore at the core of this research field, which currently focuses on pervasive adaptation. That is, the H.264/SVC stream is automatically adapted based on usage environment descriptions (e.g., client capabilities or network condition) without any user interaction. The aim is to shield the user from the adaptation process and to simply provide him with the best QoE possible given the current usage environment [3][4][5][6].

In this paper we propose a mechanism that makes it possible for the user to manually adjust the automatic adaptation process. We argue that the automatic adaptation may not always be optimal and that such a manual adjustment can help to optimize the user's QoE.

Section 2 briefly introduces the state of the art and consequently motivates our work by describing some application scenarios. Section 3 describes our approach in detail. Section 4 provides an evaluation of our approach. Section 5 gives an outlook to future work and concludes this paper

EuroITV2010, June 9th-11th 2010, Tampere, FINLAND.

2. STATE OF THE ART AND MOTIVATION

Automatic adaptation relies on knowledge of 1) the current usage environment, e.g., the current network condition or the client capabilities and 2) content characteristics, e.g., the genre. For example, if an adaptation node is aware of the currently available bandwidth, it can select the optimal enhancement layer configuration to utilize it. If this adaptation node is additionally aware of, e.g., the display resolution of the client, it can restrict the allowed enhancement layer configurations to only these which do not degrade the spatial resolution of the video below the display resolution of the client. Further optimization of the enhancement layer configuration can be performed based on content characteristics, e.g., it is not ideal to reduce the temporal resolution of a fast-moving action movie. Taking the optimal adaptation decision based on all available usage environment descriptions and content characteristics can be formulated as a mathematical optimization problem as described in [7].

Given very complete information on usage environment and content characteristics, the automatic adaptation can provide a good QoE. However, we argue that even if such complete usage environment and content characteristics information is available, which is usually not the case, the user itself must be taken into consideration in order to increase the QoE. In the following we list some cases where automatic adaptation may benefit from manual adjustments by the user:

- Incomplete usage environment information. For example, YouTube still lets the user select the spatial resolution of the chosen video.

- Incomplete content characteristics. In particular in live streams the content characteristics are usually not available.
- Interest of the user in a specific content. This can often change dynamically, e.g., during a news report.
- If a user's mobile data contract is limited to a certain data amount per month, the user will want to decide how much bandwidth to utilize for a video stream based on, e.g., his availability for the rest of the month, since he may be out of country.
- The viewing environment of the user, e.g., if the user wants to showcase a video streaming service to a friend, the user will most likely choose a higher quality.

It must be noted that some of the above cases might be expressed as *user preferences*, *user characteristics* or *natural environment characteristics* as a part of the usage environment description [8]. However, we believe that these are too multifaceted (as the cases above show) to be predefined. Additional challenges, such as privacy concerns, would also arise in this context, particularly when the adaptation decision is taken outside of the user's premises.

3. OUR APPROACH

Our approach, as introduced above, is not meant to replace automatic adaptation, but rather to enhance it by enabling the user to steer it. We therefore introduce two steering parameters:

1. Layer drop priority.
2. Minimum number of enhancement layers for each scalability dimension.

The layer drop priority is expressed as an ordered relation of the scalability dimensions D (for spatial resolution), Q (for fidelity) and T (for temporal resolution). This way the client is able to specify the requirements through the dropping priority of the scalability dimensions. Optionally, minimum values can be set for each of the scalability dimensions. Note that the *priority id* which is defined as a header field in H.264/SVC [1] has similar semantics, but cannot be used for expressing user preferences, since it is bound to the content.

Besides our steering parameters, we require a usage environment constraint to trigger the adaptation. Note that below we focus on the available bandwidth, however this could in theory be replaced by any other constraint which may trigger an adaptation.

The algorithm for the DQT selection, based on the steering parameters, is defined using pseudo code as follows:

Listing 1: Algorithm Pseudo-Code

```
Sort scalability dimensions according to drop priority,
from highest to lowest;

while (available bandwidth <
    bitrate of current enhancement layer selection) {
 for each scalability dimension {
  if (num layers of scalability dimension >
     min layers selected by the client) {
   Start filtering the currently highest layer
     from this scalability dimension;
   Stop filtering layers from scalability
     dimensions with higher drop priority;
   SelectionFound = true;
   End for loop ;
  }
 }
 if (SelectionFound == false) {
  Perform best effort adaptation;
 }
}
```

Obviously it would make no sense to adapt the bitstream if there is enough bandwidth available. Thus, the algorithm is only triggered if the available bandwidth is smaller than the maximum bit rate of the H.264/SVC stream. In this case layers have to be dropped in order to decrease the bit rate, as indicated by the steering parameters. This is done until a configuration of enhancement layers is found which has a bit rate smaller than or equal to the available bandwidth.

If no suitable enhancement layer configuration can be found, best effort adaptation is performed, selecting the base layer in the worst case.

4. EVALUATION

In order to evaluate our approach we extended the NS-2 simulator [9] to include RTCP feedback and TFRC calculation. Additionally, the different layer selection approaches presented below were implemented. The simulation setup consists of two nodes, representing a streaming server and a streaming client connected by a link. The bidirectional wire is configured to offer a total bandwidth of 2.9 Mb/s while the H.264/SVC video is encoded with an average bit rate of 2.89 Mb/s.

TCP Friendly Rate Control (TFRC) [10] is used to calculate the available bandwidth T for a certain session as defined in Equation 1.This function gives an upper limit to the bandwidth available in terms of bytes per second. The inputs to this functions are the packet size s, the Round Trip Time (RTT) r, the loss event rate p and the TCP retransmission timeout T_{RTO}. The implementation uses a mean of all packet sizes for s and a weighted moving average of the fraction lost for p, as shown in (1).

$$T = \frac{s}{r\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \quad (1)$$

Having calculated the available bandwidth, the DQT selection algorithm is triggered. Congestion is simulated after four seconds, as can be seen in the Figures 1(b), 2(b), 3(b) and 4(b). There is a delay until the actual adaptation begins, which can be explained with the weighted mean of the fraction lost p in Equation 1.

Another possibility to trigger the DQT selection is to manually change the available bandwidth, referred to as user triggered adaptation. This situation was simulated within the interval between seconds 14 and 18.

Four different approaches are compared in the evaluation. The first approach represents a best effort adaptation which selects the layer combination fitting best into the available bandwidth. The second approach sets minimal values for each scalability dimension D, Q and T, but no layer drop priority. The third approach supports layer drop priority but no minimal values. Finally the fourth approach corresponds to the one proposed in this paper, including minimal values and layer drop priority.

The video used in our simulations is the City MPEG reference video sequence with 2 spatial, 2 quality and 4 temporal enhancement values. The highest layer corresponds to 4CIF spatial resolution at 30 FPS with just one quality enhancement. For both lower spatial resolutions (CIF and QCIF) a second quality enhancement was encoded.

For the evaluation the layer drop priority has been set to $D > Q > T$, the minimum values chosen were D=1, Q=1, T=2. These minimum values correspond to CIF size and a frame rate of 7.5 FPS.

Figure 1 shows the results using the first approach, i.e. best effort adaptation. Figure 2 represents the second approach with minimal values for the scalability dimensions and Figure 3 corresponds to the third approach including the layer drop priority. Finally in Figure 4 one can see the results achieved by our proposed approach using minimal values and layer drop priority combined. Each figure consists of two diagrams. The left one shows the selection of DQT values by the approaches, while the right diagram depicts the available bandwidth which triggered the selection, as well as the corresponding selected bit rate.

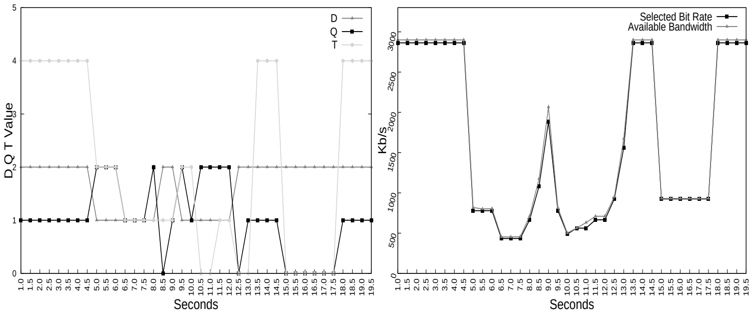
In the interval between the seconds 4 and 5.5 one can observe that the approaches with layer drop priority $D > Q > T$ (Figure 3 and Figure 4) try to keep the temporal value as high as possible. When looking at seconds 10 to 11 in Figure 3(a) and Figure 4(a), the difference between using drop priority or a combined approach becomes clearer. When in Figure 4(a), the fidelity stays higher even at the cost of loosing temporal resolution due to the defined minimums.

When looking at second 14 in the Figures the effect of the minimums becomes clearer. The result of the first approach (Figure 1) is a 4CIF slide show (1.875 FPS), while in the second approach (Figure 2) the user receives a video in CIF resolution, at high quality and with 15 frames per second. The algorithms with layer drop priority (Figure 3 and Figure 4) keep 30 FPS although at the cost of reduced quality, which corresponds to the user request. Moreover, with the third and the fourth approach less bandwidth is required compared to the first and the second approach, while perfectly matching the user preferences using the fourth approach. This shows that both layer drop priority and minimum values are relevant for an increased QoE at reasonable bandwidth utilization.

5. CONCLUSION AND FUTURE WORK

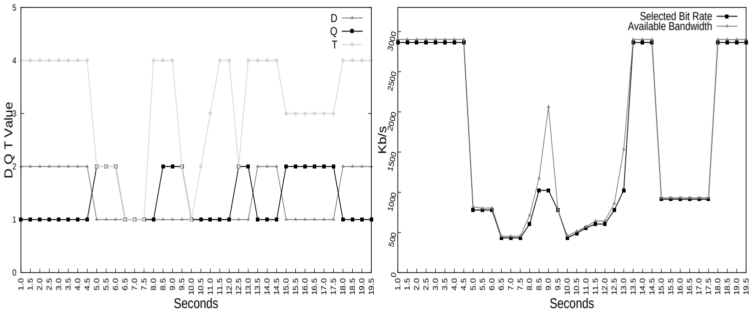
In this paper an approach for user-driven H.264/SVC video adaptation was described. After providing an introduction into the topic, we briefly presented state of the art and provided a motivation for our approach. Consequently, we presented our proposal and evaluated it. The evaluation showed the effectiveness of our approach and compared it to alternative approaches. It can be concluded that adaptation which uses our approach enables a higher QoE compared to existing adaptation mechanisms at the same bit rate.

Several items for future work can be seen. While



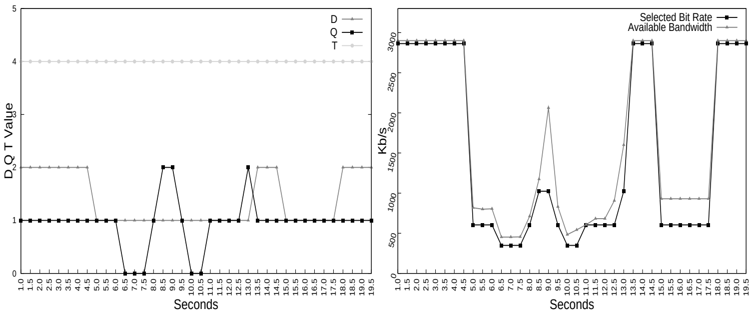
(a) DQT selection (b) Bit rate evolution

Figure 1: Best effort approach



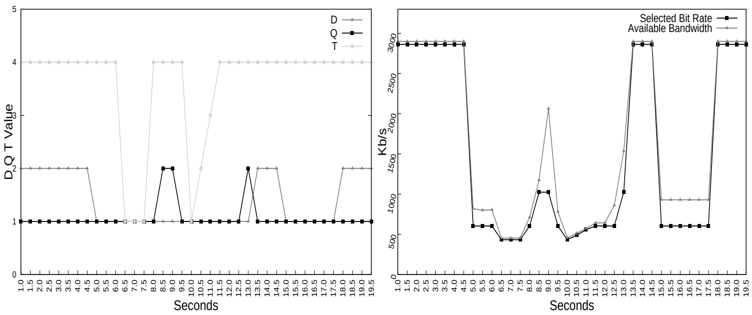
(a) DQT selection (b) Bit rate evolution

Figure 2: Minimal values approach



(a) DQT selection (b) Bit rate evolution

Figure 3: Layer drop priority approach



(a) DQT selection (b) Bit rate evolution

Figure 4: Combined approach

it is obvious that involving the user in the adaptation process results in a higher QoE, subjective tests need to be performed to be certain. An additional research item regarding the QoE is to research the user interface, i.e., how exactly can the user provide his preferences during the consumption of the video. A bad user interface could reduce the overall QoE of the user, thus negating the benefits of our approach. Finally we did not define how exactly to transmit the user preferences to an adaptation node, e.g., in an RTP streaming scenario.

6. ACKNOWLEDGEMENTS

This work is supported by the ”Ministerio de Industria, Turismo y Comercio Español (MITYC)” and the ”Österreichische Forschungsförderungsgesellschaft mbH (FFG)” in the context of the Celtic SCALNET (CP5-022) project.

7. REFERENCES

[1] T. Wiegand, G. Sullivan, H. Schwarz, and M. Wien, editors. *ISO/IEC 14496-10:2005/Amd3: Scalable Video Coding*. International Standardization Organization, 2007.

[2] T. Wiegand, J. Ohm, G. Sullivan, and A. Luthra. Special Issue on Scalable Video Coding - Standardization and Beyond. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9), September 2007.

[3] A. Hutter, P. Amon, G. Panis, E. Delfosse, M. Ransburg, and H. Hellwagner. Automatic Adaptation of Streaming Multimedia Content in a Dynamic and Distributed Environment. In *Proc. ICIP*, Genova, Italy, September 2005.

[4] I. Kofler, R. Kuschnig, and H. Hellwagner. In-network Real-time Adaptation of Scalable Content on a WiFi Router. In *Proc. 6th CCNC*, January 2009.

[5] R. Kuschnig, I. Kofler, M. Ransburg, and H. Hellwagner. Design options and comparison of in-network H.264/SVC adaptation. *Journal of Visual Communication and Image Representation*, 19(8):529–542, December 2008.

[6] C. Abhayaratne, E. Izquierdo, M. Mrak, and S. Tubaro. Special Issue on Scalable Coded Media beyond Compression. *Signal Processing: Image Communication*, 24(6), July 2009.

[7] I. Kofler, C. Timmerer, H. Hellwagner, A. Hutter, and F. Sanahuja. Efficient MPEG-21-based Adaptation Decision-Taking for Scalable Multimedia Content. In *Proc. 14th Multimedia Computing and Networking Conference*, San Jose, USA, January 2007.

[8] A. Vetro and C. Timmerer. Digital Item Adaptation: Overview of Standardization and Research Activities. *IEEE Transactions on Multimedia*, 7(3):418–426, June 2005.

[9] ns-2 Network Simulator. Technical report, June 2009. http://nsnam.isi.edu/nsnam.

[10] M. Handley, S. Floyd, J. Padhye, and J. Widmer. TCP Friendly Rate Control (TFRC): Protocol Specification. Technical report, Internet Engineering Task Force, January 2003. Standard, RFC 3448.