

User-friendly H.264/AVC for Remote Browsing

Pengpeng Ni, Damir Isovich
Department of Computer Science
Mälardalen University
Västerås, Sweden
pengpeng.ni,damir.isovich

Gerhard Fohler
Department of Computer Science
University of Kaiserslautern
Kaiserslautern, Germany
fohler@eit.uni-kl.de

ABSTRACT

With the growing popularity of variable network technologies, it is highly desirable to enable effective and quick browsing of remote multimedia content. In this paper we present a method for quick access of remote video content as an initial step towards a full digital Video Cassette Recording functionality in multimedia streaming applications such as Video-On-Demand, video broadcasting and remote video editing.

We propose a transcoding scheme for H.264/AVC video that fully utilizes the benefits of recently proposed SP- and SI-frames to facilitate user-friendly remote stream browsing and editing. The transcoding parameters can be adaptively changed and optimized to support different characteristics of H.264 video streams.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems]: Video; H.4.0 [Information Systems Applications]: General

General Terms

Algorithms, Performance, Design

Keywords

H.264/AVC, transcoding, video encoding, remote video browsing, digital VCR functionality

1. INTRODUCTION

Digital Video Cassette Recording (VCR) functionality is a key technique to support quick and user friendly browsing of multimedia contents. The set of full VCR functionality includes forward, backward, step-forward, step-backward, fast-forward, fast-backward, random access, pause, and stop operations. However, the realization of full VCR functionality over network in applications with highly compressed digital multimedia streams, e.g., MPEG, is challenging and not yet well resolved. The higher compression efficiency comes at the expense of higher computational requirement, which becomes a problem when the available decoding time for frames decreases due to a VCR trick-mode. Moreover, the MPEG stream structure imposes additional challenges. The inter-prediction technique used in MPEG allows a straightforward implementation of the forward-play function, but imposes several constraints on backward, fast-forward/backward and random access operations in terms of very high demands on network band-

width and CPU computation capacity. For example, to decode a B-frame, both a *previous* and a *later* reference frame must be transmitted and decoded first.

Video data in MPEG is organized in Group-of-Pictures (GOP), with three frame types, I, P and B. 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. One possible implementation of backward operation is to decode the entire GOP, store it in a buffer and play the frames backwards. However, this solution is not feasible in many applications due to a large memory requirement to store all the frames in a GOP, or the network delay caused by transmitting entire GOP to the client device before decoding it.

Conveniently, the current state-of-the-art of MPEG coding, the standard H.264/MPEG-4 AVC, contains several new encoding features to support quick browsing. Among them, the new frame types, SI- and SP-frames can be used to implement VCR functionality in streaming applications. One important property of SP-frames is that identical frames can be reconstructed from different reference pictures. Moreover, SP-frames have significantly better coding efficiency than I-frames while providing similar functionality [8]. Hence, they can replace I- and P-frames in applications to facilitate stream switching, splicing, error resilience and VCR like functionality [4].

Different video coding methods have been presented to address the implementation of full or part of VCR functions for MPEG compatible video streams. Several reverse-play transcoding algorithms that utilize the motion vectors to predict P- and B-frames backwards have been presented and compared in [6, 7]. However, they are based on MPEG-2 and do not suit for current state-of-art standard which takes advantage of multiple reference pictures. Additionally, the problem of extra network traffic caused by fast-playback operation is not solved as well. In [9] a solution for VCR with dual bitstreams has been presented. The idea is to offline re-encode an original MPEG video sequence in the reverse order to generate a secondary bitstream. A server process chooses the nearest I-frames from two bitstreams as start points for decoding. This solution does not enable none-drift reconstruction for bitstream switching and it has large storage requirement.

In this paper, we propose an approach for enabling the VCR functionality in H.264 streaming applications by transforming the video streams to compliant bitstreams with user friendly syntax. More specifically, we modify a subset of I- and P-frames in the original stream to primary SP-frames suitable for quick browsing, and create secondary SP/SI-frames to support different speed-ups at the decoder side. The transcoding is done in a such way that the target bit rate does not exceed a constant value which can be known in advance (e.g., the bitrate of the input stream). Given

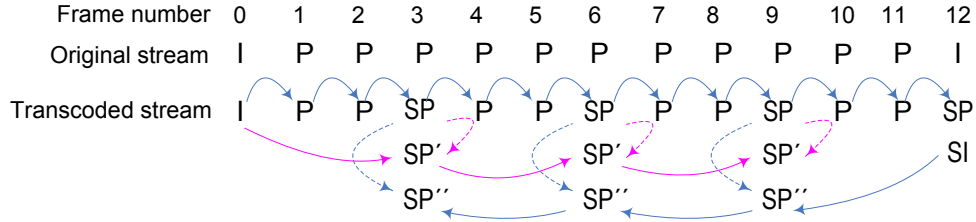


Figure 1: Encoding scheme for H.264/AVC to support digital VCR functionality.

the target bit rate, we minimize the distortion by using different settings for transcoding parameters for extra SP-frames, and apply rate-distortion theory to find the optimized combination of the encoding options.

2. MOTIVATION AND APPROACH

H.264/AVC is a flexible coding standard with a potential to meet the manifold needs of applications. One of the key features is a new frame type, S-frames [8, 12], which lead to a significant increase in coding efficiency. There are *primary* SP-frames, which use similar forward motion prediction technique as P-frames, and *secondary* SP- and SI-frames, which can be inter- and intra-coded. Each secondary frame is bounded to a primary SP-frame, and function as a substitute when functionalities like bitstream switching or random access are required.

The usage of SP/SI-frames is exploited in different application scenarios. The most discussed scenarios are error resilience and bitstream switching, see [8, 11, 13] for some examples. Our current research interest is, however, to investigate the advantage of using S-frames to facilitate the implementation of full VCR functionality over the network. We want to enable efficient jump between video frames within a single video sequence.

We propose a transcoding scheme to convert an already encoded H.264/AVC bit stream into a compliant bit stream that enables playback rate scalability. Given the attractive features of S-frames in H.264/AVC, the objective is therefore to inject the user friendly syntax in the form of SI/SP-frames while maintaining low bit rate and achieving highest quality possible. The modification is done by converting a subset of original I and P frames in the input stream into a set of primary SP-frames and corresponding secondary SI/SP-frames. SP-frames are used for forward and backward VCR modes, while SI-frames serve as random access points.

The use of S-frames does not come without penalty. The encoding process of a S-frame introduces an additional quantization step so that the coding efficiency of a primary SP-frame will never exceed the coding efficiency of a regular P-frame. Moreover, the additional quantization parameter is in general smaller than the one used in the traditional quantization step. Therefore the size of a secondary SP-frame is usually larger than its primary SP-frame, and the SI-frame is usually larger than the corresponding I-frame, as detailed in section 4.2. Thus, the choice of the value of encoding quantization parameters implies a trade-off between compression efficiency and the file size.

There are also questions around how many primary SP-frames to be placed in the output video stream since the more SP-frames the larger quality distortions. The issue here is essentially how to determine the optimal transcoding scheme for the transcoder in the rate-distortion sense.

3. STREAM TRANSFORMATION

We perform offline stream transformation. We take an original H.264 stream and modify it to a VCR friendly format by transforming a subset of original I- and P-frames into primary SP-frames. For each primary frame, SP , we generate two secondary frames SP-frames, SP' and SP'' , or one secondary SI' frame, see figure 1. The two secondary SP-frames are referenced by the nearest S-frames on their left and right side, while the secondary SI-frames do not use any references. The coding dependency between frames is illustrated by the arrows where the dashed arrows imply the coding bound between a primary and a secondary SP-frame. For the illustration simplicity, we neglect the B-frames.

The two secondary SP-frames are used to support fast-forward and backward playback modes. Assume, for example, a backward operation request on a remote stream (e.g., the one from figure 1), as depicted in figure 2. Suppose that the currently decoded frame is P_8 . Once when P_8 has been decoded and displayed, its reference frame P_7 is discarded from the frame buffers, which means that it must be decoded once again. In the original stream, the only way to decode P_7 is to decode the entire chain of reference frames starting with the I-frame (P_7 needs P_6 which needs P_5 which needs P_4 and so on). In the modified stream, however, P_7 can be decoded from the primary SP_6 frame, or from some of its secondary frames SP'_6 or SP''_6 . So, the client sends the video segment $\{SP_9, SP''_6, P_7\}$, which is clearly less than 7 frames in the original stream.

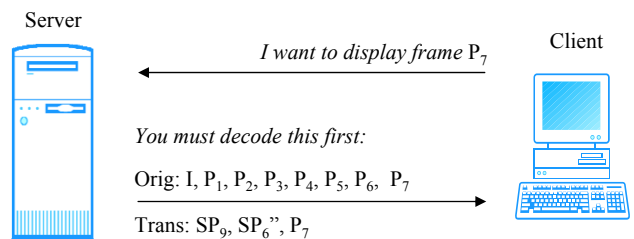


Figure 2: Remote backward playback.

Consider another example where the current frame is P_1 and fast-forward playback with speed-up factor 9 is requested. The next frame to decode is P_{10} . In the original stream, we need to transmit 9 frames, i.e., $P_2..P_{10}$. In the transcoded stream, we can send the segment $\{P_2, SP_3, SP'_6, SP'_9, P_{10}\}$, i.e., 5 frames instead of 9. Actually, we can do it even better, with only 3 frames $\{SI_{12}, SP''_9, P_{10}\}$. However, SI_{12} is intra-coded, so the second segment could have larger bit size than the first one but there is a choice on the server side which segment to send. The benefit of using SP-frames became even more obvious with higher speed-ups and larger original GOPs.

4. ADAPTIVE TRANSCODING SCHEME

An absolute optimized stream transformation is hard to find since various encoding schemes can show various efficiency at different bit rates and with different scene content. However, it is useful to be able to adaptively change the option settings of the transcoding scheme with respect to the characteristic of different video streams. We define the settings of our transcoding scheme by the variations in proportion of S-frames and using different quantization parameters for frame encoding.

4.1 Proportion of S-frames

The proportion of S-frames decides the amount of S-frames added into a video stream. Let N denote the number of frames between two SI-frames, and n the number of frames between two adjacent primary SP-frames. Then, assuming the primary SP-frames are uniformly distributed in the video stream, the proportion of S-frames is given by $\rho = n/N$. For example, in figure 1, $\rho = 0.5$ when $n = 3$ and $N = 12$. By varying the proportion of S-frames in the transcoding scheme, we can achieve different distribution of S-frames in a transcoded stream.

In our current transcoding scheme, all I-frames in a video sequence, except the first one, are replaced by primary SP-frames. Since an SP-frame is generally smaller than an I-frame, bits can be saved and evenly allocated to the other inter-predicted frames. Notice that the transformation of an I-frame to a primary SP-frame make most sense when the predecessors P-frame is similar to the I-frame, which usually is the case in a video sequence. The difference between the last frame in a sequence and the first frame in the next sequence is most likely to be quite large, hence we do not transform the first I frames in video sequences. Original I-frames are also needed for the error resilience.

Bounded to the primary SP-frames, secondary SI-frames can be adaptively generated based on the stream demands and random access requirement. The original GOP sizes are not necessarily preserved since SI-frames are generated both from original I- and P-frames.

We also found that the number of frames between two adjacent primary SP-frames, n , has major effect to the decoding time of backward playback operation. The larger n , the longer the decoding time. For example, if the frame P_8 in the figure 1 is followed by additional P-frames, then all of them must be sent by the server and decoded together with the video segment $\{SP_9, SP_6'', P_7\}$.

However, not all P-frames can be transcoded to SP-frames, partly because of the larger storage space required, and partly because of the quality loss due to the different settings for the quantization parameters for encoding. Obviously, there is a trade-off between the number of transcoded P-frames and the decoding latency. Combining the choice of suitable value for n with multiple reference picture control, the decoding time jitter of backward playback can be minimized. This is, though, a part of our ongoing research and it is not reported in this paper.

4.2 Quantization parameters

In our approach, we can also vary the quantization parameters. We illustrate briefly the encoding process in Figure 3 (abstracted from the design introduced in [12]).

K_{pred1} and K_{pred2} are the prediction coefficients used for inter prediction. They are generated separately using different reference pictures (note that the coefficients are already transformed prior inter-prediction). For example, to encode primary SP-frame, K_{pred1} is subtracted from the original video frame to get the difference between the predicted picture and its reference picture. The

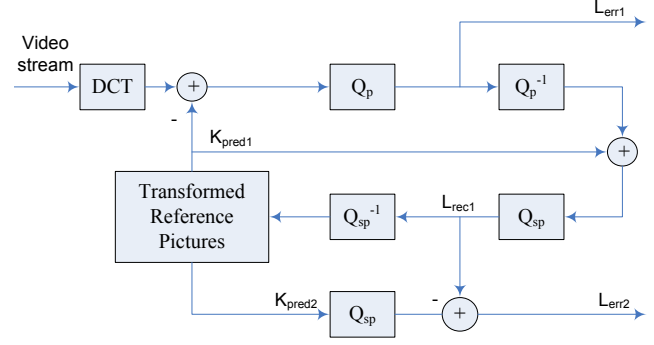


Figure 3: Simplified SP-frame Encoder.

difference is then quantized using the parameter Q_p . The quantization step will cause some quality loss. We denote the quantized error level by L_{err1} which is eventually entropy encoded to form the compressed bitstream. The encoding process of a primary SP-frame is similar to that of a traditional P-frame. However, the difference is made by an additional quantization/dequantization step in the reconstruction procedure (Q_{sp}, Q_{sp}^{-1} in the figure 3). L_{rec1} denotes the quantized coefficients of the video frame reconstructed from primary SP frame. It is compared with K_{pred2} to generate L_{err2} which represents the secondary SP-frame prior entropy encoding. Note that K_{pred2} needs also to be quantized by the same quantization parameter Q_{sp} before the subtraction. It is obvious that the identical reconstruction can be obtained again at the decoder, since:

$$L_{rec1} = L_{err2} + Q_{sp}(K_{pred2}) = Q_{sp}(Q_p^{-1}(L_{err1}) + K_{pred1})$$

Finally, the corresponding SI-frame can be generated from L_{rec1} using intra prediction techniques.

Looking through the encoding process, we notice that the additional quantization may not affect direct the quality of the primary SP-frame, and the identical reconstruction of the primary SP-frame using secondary SP-frame is also guaranteed regardless the value of Q_{sp} . Moreover, the coding efficiency of primary SP-frame will never exceed the coding efficiency of the corresponding P-frame. Also, in general, the quantization parameter Q_p is larger than Q_{sp} so that the size of the secondary SP-frame is usually larger than the primary SP-frame, and the secondary SI-frame is usually larger than the regular I-frame, see [14].

However, the extra quantization affects the encoding of the following video sequence. A coarser Q_{sp} may lead to poorer prediction for the later encoded video frames, while a finer Q_{sp} gives larger secondary SP- and SI-frame size.

In summary, the two quantization parameters Q_p and Q_{sp} influence mutually the rate distortion performance overall the entire video stream. Finer Q_p and Q_{sp} means better quality but poorer compression efficiency. These parameters can be adaptively changed so that bits allocated for a GOP can be evenly or strategically distributed among video segments separated by primary SP-frames.

4.3 Transcoding optimization

The transcoding parameters have strong influence to the stream's rate-distortion performance. We apply Lagrangian optimization technique to find the optimized combination of these encoding options, which is widely used by industries due to its effectiveness, conceptual simplicity, and capability of combining numbers of parameter settings, see [2, 5, 10]. The basic idea is to minimize the

Lagrangian cost formulation given by $J = D + \lambda R$, where D represents the distortion term which can be quantified by some distortion measurement method, for instance mean square error(MSE). R is the average rate term which is measured in unit of bits/second. The Lagrange multiplier λ is a non-negative real number which determines the weight assigned to rate term against distortion term. Its value is decided according to application's different specification. In geometry, to minimize the cost formulation J , the value of λ corresponds to the slope of rate-distortion function (RDF) curve generated specifically for a video stream.

We do our transcoding in an optimal fashion with the guidance of Lagrangian optimization theory. At first, we generate three RDFs for video streams. Each RDF is corresponding to the changes of one encoding parameter while keeping the other two static. Secondly, we look for the optimal operating points in the rate-distortion diagrams. The slopes at these optimal points for each RDF curves should be equal, while the sum of the bit rates corresponding to these operating points in the RDF diagrams should be minimized. We use these optimal points to determine the optimal value for our transcoding parameters.

5. ONGOING EXPERIMENTS

Currently, we are performing experiments to verify the proposed approach. We use the H.264/AVC reference software JM 10.2 [1] and the SP-frame encoder written by Eric Setton of Stanford University [11]. We generate reference video sequences conformed to H.264 baseline profile using standard uncompressed video test sequences at CIF format such as "Paris" and "Tempete". The GOP size of each video sequence is set to 16. By changing quantization parameter for I and P frames from 16 to 36 at interval of 4, we obtain reference video sequences with different content and at different average bitrates.

The test sequences are encoded again to generate video sequences containing primary SP-frames. Only the first frames in each sequence is encoded as an I-frame while other frames are encoded as either P- or primary SP frames. The primary SP frames are encoded periodically at intervals of 2,4,6,8 frames. In these SP-frames Q_{sp} is set to be equal to Q_p whose value varies with increment from -3 to 3.

We generate RDFs corresponding to the variation of encoding parameters and apply Lagrangian technique to find optimized encoding parameters based on RDF diagrams. The final optimized video sequences with secondary SP- and SI-frames will be compared with the reference video sequence in compression efficiency, file size and decoding time. The complete analysis results will be reported in our forthcoming publications.

6. CONCLUSIONS

In this paper we presented a transcoding scheme for H.264/AVC video to support user-friendly real-time browsing. A sub-set of original I- and P-frames is transformed into SI/SP-frames suitable for quick browsing of local and remote video content. The transcoding parameters can be adaptively changed and we apply rate-distortion theory to find the optimized combination of the encoding options.

The work presented here is a first step towards a full digital VCR functionality through a network. Currently, we are implementing the proposed approach and evaluating the trade-offs between user-friendly browsing and the resource usage.

Moreover, we are investigating how shot change detection can be incorporated in our transcoding method to improve the distribution of non-transformed I frames.

Furthermore, we want to minimize the jitter/latency of the decoding. For this purpose, we are looking into Multiple Reference Control and Decoded Picture Buffer of H.264. With adaptive control, the memory usability can be improved which in return decreases decoding latency.

Frame dropping technique can also be used to reduce the computational requirements. In our previous work [3], we proposed a quality-aware frame selection algorithm for MPEG-2, which we are now extending to support the S-frames.

7. REFERENCES

- [1] H.264/AVC reference software (JM 10.2). <http://iphome.hhi.de/suehring/tml/download/>.
- [2] H.Everett III. Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *IEEE transactions on circuits and systems for video technology*, 11, May 1963.
- [3] Damir Isovich, Gerhard Fohler, and Liesbeth F. Steffens. Real-time issues of MPEG-2 playout in resource constrained systems. *Journal of Embedded Computing (JEC), special issue 3*, June 2004.
- [4] ITU-T and ISO/IEC JTC 1. *Advanced Video Coding for Generic AV services*, April 2003.
- [5] Gary J.Sullivan and Thomas Wiegand. Rate-distortion optimization for video compression. *Signal Processing Magazine, IEEE*, 20, November 1998.
- [6] Susie J.We. Reversing motion vector fields. In *Image Processing, 1998. ICIP 98. Proceeding*, volume 2.
- [7] Susie J.We and Bhaskaran Vasudev. Compressed-domain reverse play of MPEG video streams. In *SPIE*, November 1998.
- [8] Marta Karczewicz and Ragip Kurceren. The SP- and SI-frames design for h.264/avc. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), July 2003.
- [9] Chia-Wen Lin, Jian Zhou, Jeongnam Youn, and Ming-Ting Sun. Mpeg video streaming with vcr functionality. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3), March 2001.
- [10] Antonio Ortega and Kannan Ramchandran. Rate-distortion methods for image and video compression. *Signal Processing Magazine, IEEE*, 15, November 1998.
- [11] Eric Setton and Bernd Girod. Video streaming with SP and SI frames. In *Proceedings VCIP,Beijing*, July 2005.
- [12] Xiaoyan Sun, Shipeng Li, Feng Wu, Jacky Shen, and Wen Gao. The improved sp frame coding technique for the JVT standard. In *ICIP 2003*.
- [13] Wai tian Tan and Gene Cheung. SP-frame selection for video streaming over burst-loss networks. In *IEEE International Symposium on Multimedia*, 2005.
- [14] X.Sun, F.Wu, S.Li, W.Gao, and Y.-Q Zhang. Improved SP coding technique. Technical report, January 2002.