



biblio.ugent.be

The UGent Institutional Repository is the electronic archiving and dissemination platform for all UGent research publications. Ghent University has implemented a mandate stipulating that all academic publications of UGent researchers should be deposited and archived in this repository. Except for items where current copyright restrictions apply, these papers are available in Open Access.

This item is the archived peer-reviewed author-version of:

Fast Transcoding for Video Delivery by Means of a Control Stream

Glenn Van Wallendael, Jan De Cock, Rik Van de Walle

Proceedings of IEEE International Conference on Image Processing (ICIP), Orlando, USA, September 2012.

To refer to or to cite this work, please use the citation to the published version:

G. Van Wallendael, J. De Cock, R. Van de Walle, "Fast Transcoding for Video Delivery by Means of a Control Stream", Proceedings of IEEE International Conference on Image Processing (ICIP), Orlando, USA, September 2012

FAST TRANSCODING FOR VIDEO DELIVERY BY MEANS OF A CONTROL STREAM

Glenn Van Wallendael, Jan De Cock, and Rik Van de Walle

Ghent University – IBBT, ELIS Department - Multimedia Lab,
Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent, Belgium

ABSTRACT

When a video delivery system provides different representations of the same sequence, solutions can be found in simulcast coding, scalable coding, and transcoding. The major downside of transcoding is the additional complexity needed to re-encode the video sequence in its new form. When one fidelity version of the video stream should be transmitted over the network, scalable coding is less efficient compared to single layer coding because of the layering overhead. Finally, simulcast encoding results in large storage requirements. In this paper, we propose an intermediate solution providing transcoding at a low complexity by the aid of control streams. We define a control stream as a regular video stream from which residual information is removed. With these control streams, the complexity of the encoding step in the transcoder can be reduced to decoder complexity. As a result of the removal of residual information, these control streams take up 62.8% less bitrate compared to simulcast coding for a test set based on High Efficiency Video Coding (HEVC). Additionally, when compared to scalable coding, an efficient single layer video stream can be provided without the 16.6% bitrate increase caused by transmitting a layered bitstream.

Index Terms— transcoding, scalable coding, HEVC, H.264/AVC

1. INTRODUCTION

When providing video services, product differentiation regarding Quality of Service (QoS) can be achieved by adapting the properties of the video to the end user's device characteristics and network restrictions. Transcoding, simulcast coding, and scalable coding are the enabling technologies for these adaptation requirements.

With a transcoding solution [1], a single source sequence is kept in the video storage and different versions are created on-the-fly upon request. These different outputted video streams can have a lower spatial resolution, a lower temporal resolution, a lower quality, or even a different compression standard. On an abstract level, transcoding solutions come in two variations, namely compressed domain and Cascaded Pixel Domain Transcoders (CPDT). Compressed domain transcoding provides a low complexity solution for adapting

video sequences, but at a quality penalty caused by open-loop modifications. With a CPDT, the video stream is decoded and re-encoded with the desired properties. This re-encoding procedure is very time consuming but can get accelerated with information from the source video stream [1]. In general, the downside of transcoding is the high computational complexity needed to create the output video stream.

When a low complexity solution is needed, simulcast coding can be considered. With this approach, every provided variation of the original video stream is stored separately on the video storage. Upon request, all different versions are available and can be provided to the end user without additional complexity. As a downside, a lot of storage is needed to retain all the different versions of the original video stream.

Finally, there is the option for scalable storage of the different versions of the video stream. With a scalable compression standard like the Scalable Video Coding (SVC) [2] extension of H.264/AVC (Advanced Video Coding), different variations of a video stream are stored in a single video stream in a layered way. In an SVC video stream, a Base Layer (BL), having the lowest fidelity, is improved with Enhancement layers (EL) with increasing bitrate and fidelity. Discarding these ELs from the scalable bitstream only requires low complexity operations on that bitstream resulting in lower fidelity versions of the video. Storing a scalable video stream is more efficient than simulcast storage because with scalable coding the low quality representations can be used for more efficient compression of the ELs. On the contrary, transmission of a scalable video stream is less efficient than transmission of a single layer video because layering a video stream results in an additional bitrate cost.

In this paper, we propose a transcoding solution requiring less computational power than a CPDT transcoder, requiring less storage than a simulcast solution, and providing an efficient single layer video stream to the end user. To explain the proposed transcoding scheme, first, the time consuming process of video encoding is explained with a focus on the High Efficiency Video Coding (HEVC) standard in Section 2. Second, the proposed transcoding solution is elaborated on in Section 3. Results applying the proposed technique on H.264/AVC and HEVC can be found in Section 4. Finally, conclusions are given in Section 5.

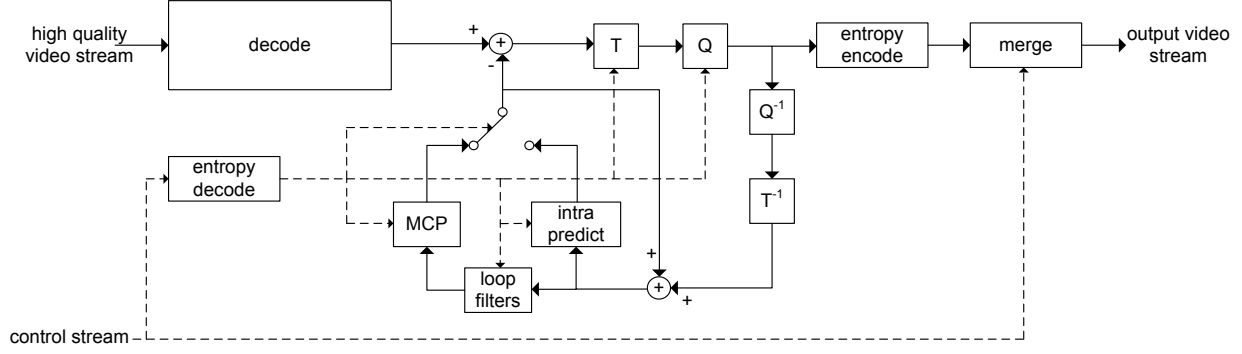


Fig. 1. Proposed fast decoder-encoder cascade by means of a control stream. (T: transform; Q: quantization; MCP: motion compensated prediction)

2. VIDEO ENCODER COMPLEXITY

When converting a video stream to a different form by means of a cascade of decoder and encoder, the encoding process is by far the most complex part of the system. This complexity is caused by the large variety of options the encoder can utilize when encoding a certain area in the image. Although the encoder is free to use a very low complexity subpart of a specified codec, making use of the full potential of a codec will eventually result in more efficient compression.

To explain the complexity of a block based video encoder, terminology as used in HEVC [3] will be used. Although H.264/AVC is also considered in the results section, it is not chosen as the basis for the encoder description because it is more restricted than HEVC. In general, the principles behind H.264/AVC encoding and HEVC encoding are resembling.

In HEVC, a slice of an image is divided in Largest Coding Units (LCU) of 64x64 pixels which are processed in a raster scan order. Each LCU can then be divided in Coding Units (CU) according to a quadtree structure. Three levels of quadtree division can be applied until a minimum CU size of 8x8 is reached. Any combination of quadtree splitting is allowed, resulting in 83522 possible quadtree structures. For every leaf CU of the quadtree, an intra or inter prediction type is determined.

The leaf CU is then filled up with Prediction Units (PU). For an intra CU, the CU can be regarded as one PU or split in four smaller PUs. Depending on the size of the PU, up to 35 intra prediction modes can be chosen. On the other hand, when the leaf CU is an inter coded CU, eight different partitioning structures are available. On these square or rectangular partitions Motion Compensated Prediction (MCP) is applied. For a typical encoder, the motion estimation process has the highest impact on encoding time. First of all, there is the flexibility to choose from different pictures to reference to. Afterwards, for each PU, motion vectors resulting in a minimal prediction error must be searched for in this set of reference pictures.

Next, in HEVC, there is the compression efficient option to merge a PU with a neighboring PU. The prediction information of this neighboring PU is then applied on the merged PU. Back on the CU level, there is also a compression efficient option called skip. As an exception, for a skipped CU, the difference between the original CU and the predicted CU, which is called the residual information is not transformed and signaled. For every other kind of CU, the residual information is transformed and signaled on the Transform Unit (TU) level.

Starting on the leaf CU size, a recursive partitioning in smaller TUs depending on the PU shape can be applied. In the leaves of the TU tree, there is the option to change the quantization for each TU followed by the transformed and quantized residual information.

After predicting, transforming, and quantizing the picture, three loop filters are applied on the picture. First, a deblocking filter tries to mask blocking artifacts created by the quantization process. For the deblocking filter, no decisions regarding compression efficiency need to be made by the encoder. On the other hand, for the Sample Adaptive Offset (SAO) filter the encoder again has to decide on an efficient quadtree partitioning. For SAO filtering, an SAO type and offset are decided and signaled by the encoder. The last filter that can be applied is the Adaptive Loop Filter (ALF). First, the encoder can calculate up to 16 different filters that can be used. For every CU, the encoder needs to decide which filter to use to get the best Rate-Distortion (RD) result. With current HEVC encoder implementations, the ALF filter takes up the largest amount of processing power from all filters.

3. PROPOSED TRANSCODING

For the proposed transcoding scheme to be practically applicable, we assume that different predefined versions of the same video content are made available and that there is one high quality version from which the lower fidelity video streams are derived.

Table 1. BD bitrate difference of the proposed system compared to different competing technologies.

Sequence	simulcast one stream	simulcast both streams	scalable coding
Kimono	-69.6%	-20.7%	3.6%
ParkScene	-60.8%	-16.7%	4.4%
Cactus	-62.4%	-16.9%	4.6%
BasketballDrive	-60.4%	-16.0%	9.0%
BQTerrace	-71.5%	-17.2%	1.7%
BasketballDrill	-55.3%	-15.5%	8.5%
BQMall	-55.4%	-15.7%	6.6%
PartyScene	-65.9%	-18.1%	2.8%
RaceHorses	-60.4%	-15.0%	9.6%
BasketballPass	-56.1%	-15.2%	9.3%
BQSquare	-66.2%	-18.7%	0.5%
BlowingBubbles	-63.3%	-17.0%	2.0%
RaceHorses	-52.7%	-13.5%	9.9%
ChinaSpeed	-62.8%	-18.6%	6.0%
SlideEditing	-81.8%	-32.7%	7.1%
SlideShow	-60.4%	-20.2%	11.3%
Avg. HEVC	-62.8%	-17.6%	6.1%
Avg. H.264/AVC	-62.7%	-17.7%	4.5%

In general, the system behaves as a cascade of a decoder and encoder with drastic encoder complexity reduction applied. For this complexity reduction to have no impact on encoder compression efficiency, control streams are created for each low fidelity video stream. The information in the control stream contains all the decisions the encoder has to make.

To make a different version of a video stream available, a control stream needs to be made. This process starts by creating the video stream as it is supposed to be output by the system. All intensive encoder processing happens at this stage and only needs to be done once for every provided version. As a result, the simulcast version of the video stream is obtained. To reduce this video stream in size, all residual information contained on the TU level is removed from the video stream and a control stream is obtained. This control stream is then stored together with the high quality video stream where it is derived from.

When a low fidelity version of the video stream is requested from the system, the process as illustrated in Fig. 1 is performed. The high quality video stream needs to be decoded to have access to the original pixels from which the low fidelity versions are derived. Then, the control stream is entropy decoded making available all decision information to the encoder. The CU quadtree can be read out of this control stream reducing the possible quadtree structures to a single option. Then, for every CU, the control stream provides skip and prediction type information. The PU partitioning structure and the motion information is made available on the PU level of the control stream. Consequently, no complex motion estimation procedure needs to be carried out. On a TU level, information about transform level splitting is provided

together with Quantization Parameter (QP) changes. The only information that needs to be added back to the control stream to make it a complete video stream again is residual information. This residual information results from the encoding loop and will be merged back in the control stream after entropy encoding.

To close the prediction loop entirely, loop filtering still needs to be applied. All loop filters will end up having the complexity as needed by a decoder. The filter coefficients of the ALF filter and the offsets of the SAO filter are available from the control stream resulting in a low complexity processing of these tools.

With all the encoder complexity reductions as provided by the control stream, the encoder complexity is made similar to that of a decoder. The only difference between the low complexity encoder with a regular decoder is that on the TU level, the residual is not decoded from the bitstream, but created with the high quality reference stream.

When creating a control stream encoder and transcoder, it must be taken into account that encoder quantization tools like Rate Distortion Optimized Quantization (RDOQ) must be of an identical algorithm in both the control stream encoder and the encoder part of the transcoder. Otherwise, a different residual signal would be created resulting in quality loss of the video stream.

4. RESULTS

The proposed system is evaluated on systems based on HEVC and H.264/AVC. The HEVC system is derived from the HEVC Reference Model version 4.0 [4] and for H.264/AVC the JSVM Reference Model version 9.19.14 [5] is adapted. For both compression technologies, a comparison is made with simulcast coding and scalable coding. Because a scalable version of HEVC is not defined yet, a scalable extension of HEVC is created similar in principle to the Multiview Video Coding (MVC) extension of H.264/AVC. For the H.264/AVC based system, scalable coding is realized by means of Coarse Grain Scalable (CGS) coding of the SVC extension.

To simulate a video delivery scenario, different parameters are applied on the video stream. A Group Of Pictures (GOP) size of eight is chosen with hierarchical coding within the GOP. Furthermore, a random access period of 32 is set to enable random access at regular intervals. To obtain realistic PSNR values in the range of 30 dB to 40 dB, QP parameters of 22, 27, 32, and 37 are chosen. In both H.264/AVC and HEVC, CABAC is selected as the entropy encoder. With regard to loop filtering, H.264/AVC only allows to make use of the deblocking filter. With HEVC, the deblocking filter, SAO filter, and ALF filter are enabled for the tests. For the alternative versions of the original video stream, the QP is increased with five resulting in approximately half the bitrate. So, for every sequence and every QP, both a high quality video

Table 2. BD bitrate difference of competing technologies compared to a transcoding scenario storing only one high fidelity video stream.

Sequence	scalable coding	proposed	simulcast
Kimono	14.3%	18.3%	49.2%
ParkScene	16.1%	21.1%	45.4%
Cactus	17.9%	23.1%	48.1%
BasketballDrive	14.7%	24.8%	48.7%
BQTerrace	13.7%	13.5%	37.1%
BasketballDrill	18.0%	27.7%	51.1%
BQMall	20.3%	28.1%	52.0%
PartyScene	17.9%	21.1%	47.8%
RaceHorses	14.0%	24.8%	46.8%
BasketballPass	17.2%	27.9%	51.0%
BQSquare	21.1%	21.4%	49.5%
BlowingBubbles	20.1%	22.4%	47.4%
RaceHorses	17.6%	29.1%	49.2%
ChinaSpeed	15.3%	22.1%	50.0%
SlideEditing	9.0%	16.4%	73.0%
SlideShow	17.9%	30.7%	63.7%
Avg. HEVC	16.6%	24.0%	50.7%
Avg H.264/AVC	17.5%	22.3%	48.5%

stream and an low fidelity alternative is created. Bitrate and PSNR differences are summarized over all QPs by means of the Bjøntegaard Delta (BD) [6] metric.

A comparison is made between a regular video stream as stored in a simulcast scenario and a control stream as created in the proposed solution. The results of this comparison for HEVC based compression can be found in the first column of Table 1. On average, 62.8% less bitrate is needed to store a control stream instead of the full video stream. When also considering the storage needed for the high quality reference video stream, the total bitrate saving compared to simulcast corresponds to 17.6% as indicated in the second column. From these results it can be deduced that adding more alternative versions of the same video stream will improve this bitrate gain. In the last column, the results of a comparison of the sum of both the reference stream and the control stream compared to a scalable stream is given. The proposed solution requires 6% more bitrate on the storage system compared to an MVC-like scalable solution. Under the tested conditions, storage of a scalable video stream may be more efficient, but when transmitting the scalable video on the network, overhead caused by layering the video stream must be taken into account. The cost resulting from layering the video stream will be discussed later.

When evaluating the same comparisons for an implementation based on H.264/AVC and SVC, similar results are found. Consequently, only average results are summarized in the bottom row of Table 1.

Finally, a comparison is made between transcoding and scalable coding, the proposed solution, and simulcast coding respectively. With the transcoding scenario, there is only one

high quality video stream stored on the system. Storing two variations as a scalable video stream increases the total bitrate with 16.6% as can be noted in the first column of Table 2. As already mentioned in the previous comparisons, this bitrate increase will also be present on the network when the full quality video stream is provided. Although scalable coding is 6.1% more efficient to store, transferring the video over the network requires 16.6% more bandwidth. It must be noted that with the transmission of a scalable bitstream there is still the possibility to adapt the bitstream in the network, which is not possible with single layer bitstreams as provided by other solutions. In the second column of Table 2, it can be observed that with the proposed solution, 24.0% bitrate increase is measured compared to the solution with a single video stream. The simulcast configuration results in 50.7% bitrate increase on the storage system.

5. CONCLUSION

When different versions of a video sequence are made available with a video delivery system, three different options exist, namely simulcast coding, scalable coding, and transcoding. In this paper, we provide a fast transcoding solution reducing the bitrate of every additional video representation with 62.8% when compared to simulcast storage. When only looking at storage, the proposed solution is 6.1% worse than scalable coding, but on the transmission side, 16.6% less bitrate is needed on the network caused by the difference between single layer coding and scalable coding. Finally, compared to a transcoding solution consisting of a decoder-encoder cascade, the complexity needed by the system is comparable with the complexity needed by two decoders. Overall, when more different versions of a video stream are needed, the system will become more efficient regarding storage and computational complexity.

6. REFERENCES

- [1] A. Vetro, C. Christopoulos, and H.F. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, Mar. 2003.
- [2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding extension of the H.264/AVC standard," vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
- [3] B. Bross, W.-J. Han, J.-R. Ohm, G.J. Sullivan, and T. Wiegand, "WD4: Working Draft 4 of High-Efficiency Video Coding," *JCTVC-F803*, Torino, Italy, Jul. 2011.
- [4] K. McCann, B. Bross, S.-i. Sekiguchi, W.-J. Han, "HM4: High Efficiency Video Coding (HEVC) Test Model 4 Encoder Description," *JCTVC-F802*, Torino, Italy, Jul. 2011.
- [5] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Doc. JVT-AB203: Draft reference software for SVC," Tech. Rep., MPEG / ITU-T, Jul. 2008.
- [6] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *document VCEG-M33 of ITU-T Video Coding Experts Group (VCEG)*, Apr. 2001.