Low-complexity video content adaptation for legacy user equipment

Sandro Moiron[1], Sérgio Faria[1,2], Pedro Assunção[1,2], Vitor Silva[1,3], Antonio Navarro[1,4]

[1]Instituto de Telecomunicações, Portugal; [2]Instituto Politécnico de Leiria, ESTG, Portugal;
[3]Universidade de Coimbra, DEEC, Portugal; [4]Universidade de Aveiro, DET, Portugal.
{sandro.moiron, sergio.faria, amado, vitor}@co.it.pt, navarro@av.it.pt

ABSTRACT

This paper deals with the adaptation problem raised by the fact that H.264/AVC standard is rapidly being adopted by content and service providers while at the user side MPEG-2 equipment will last much longer. A transcoding scheme is proposed for adapting compressed video content to legacy user equipment in the near-future technology transition period. The proposed transcoder is shown to be computationally efficient and exhibits good performance when compared with single step encoding at the same output format. Possible applications include additional video content adaptation modules with transcoding functionality in either home gateways or adaptation devices, in wireless content distribution scenarios where H.264/AVC is preferred because of its higher efficiency.

Keywords

Content adaptation, video coding

1. INTRODUCTION

Multimedia services and applications mostly rely on compressed video where the H.264/AVC standard [1] plays a major role due to its higher coding efficiency when compared with other video coding standards [2]. At the same video quality H.264/AVC encoding achieves 2 to 3 times lower bit rate than its counterpart MPEG-2 [3]. For this reason it is the preferred compression format for both content and service providers since lower storage capacity and network bandwidth is required for multimedia services delivery. However, the MPEG-2 standard is nowadays the most popular format supported by end-user equipment because it is on the market for quite a long time and it is also very cheap technology. Although MPEG-2 user equipment might be replaced in the future by its H.264/AVC equivalent, it is expected that during a relatively long transition period both technologies will co-exist giving rise to interoperability problems [4]. This can also be a slowdown factor for service providers trying to increase market penetration and revenues by attracting more service users. Therefore content adaptation between these two coding standards will be a crucial issue within the technology transition period be-

cause both content and service providers will migrate to the new technology much faster than users.

A general overview of a possible application scenario is depicted in Figure 1. From the service provider down to the user premises, H.264/AVC coded video is used because low bandwidth is a requirement as pointed out above. Note that this is even more important in wireless access networks such as WiMAX given as an example in the Figure 1. Then at the service gateway an adaptation module comprised of a video transcoder like the one proposed in this paper performs the necessary conversion for MPEG-2 format in order to make it compatible with the user equipment. It should be pointed out that within the user premises the low bandwidth requirement (i.e., high coding efficiency) is not as stringent as in the access network, which allows the bit rate of the H.264/AVC incoming stream to increase in order to keep the signal quality after conversion to MPEG-2.

The following sections of the paper describe a low complexity transcoding scheme for adaptation of H.264/AVC compressed video to MPEG-2 legacy user equipment. It is based on a cascaded H.264/AVC decoder and MPEG-2 encoder and a coding mode conversion module which converts the coding modes of H.264/AVC into MPEG-2 format, by using low computational complexity techniques. Overall the computational complexity of the proposed transcoder is much lower than the corresponding cascade of decoder-encoder while the signal quality remains roughly the same because of the fast conversion methods devised for this purpose.
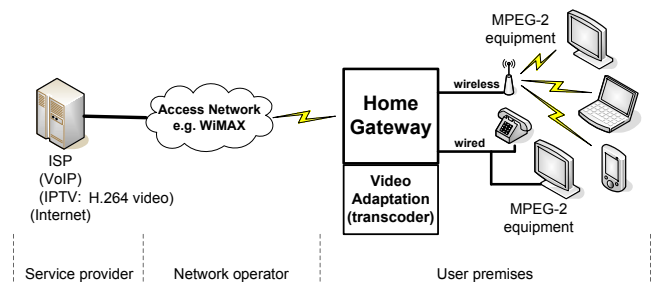

Figure 1: Application Scenario.

2. SYSTEM ARCHITECTURE

Most techniques used to compress video signals, in particular many of those adopted in the H.264/AVC standard, require very complex operations and huge time consuming. Thus, it usually needs hardware circuits or high performance processors to achieve the desired performance. If one can alleviate the computational complexity of such systems, then
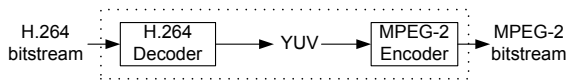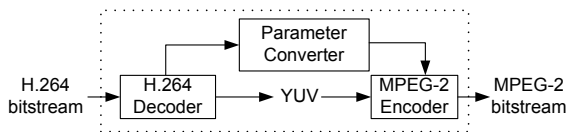
Figure 2: Cascade Transcoder.

Figure 3: Proposed Transcoder.

Figure 4: Transcoding Architecture.

their implementation can benefit from it namely in price and power consumption. In the case of conversion of a compressed video stream into another format, the computational complexity can be further reduced by taking advantage of the embedded information produced by the first source encoder.

The system architecture used to perform the video conversion is of major importance role since it has a significant impact on the quality of transcoded video, as well as in the CPU and memory resources necessary to fulfill the operational requirements. The most simple architecture is composed by a cascaded decoder/encoder, such as the one shown in Figure 2, where conversion is performed by decoding the input signal up to the pixel domain and encoding it again. Although such an architecture is able to maintain the image quality of the input bitstream, it is computationally inefficient, since it decodes the source bitstream into a uncompressed raw format and then fully encodes it again without taking advantage of the coding information available in the original stream. The encoding step includes motion estimation which implicitly has a non-desirable high computational complexity associated with it, contributing with a significant penalty in the overall computational complexity performance of the transcoder.

Since H.264/AVC and MPEG-2 standards share several features, like interframe block based motion estimation and intraframe transform coding, the proposed architecture, illustrated in figure 3, exploits such similarities in order to enhance the system performance. By reusing several parameters of the embedded information in the H.264/AVC bitstream, such a architecture strongly improves its performance in terms of computational complexity when compared with the reference cascade transcoder. Based on the classical cascade approach, the proposed transcoder architecture is composed by an H.264/AVC, decoder JM10.2 [5], followed by a MPEG-2 encoder, v1.2 from the *MPEG Software Simulation Group* [6].

A more detailed view of the proposed transcoder is shown in figure 4, where each functional module is displayed. The proposed architecture comprises an H.264/AVC decoder,a MPEG-2 encoder and a "Conversion Module" between them. This new module is the core of the system which introduces new functions to exploit the information embedded in H.264/AVC bitstreams by extracting those relevant coding parameters that allow smart simplification of the MPEG-2 encoding routines. This module works in parallel with the classical cascade transcoder, interacting with the encoder when necessary.

Among the several functional modules that compose the MPEG-2 encoder, motion estimation reveals to demand more computation than others, and therefore it is the main target for optimisation. Thus, by reducing its complexity there
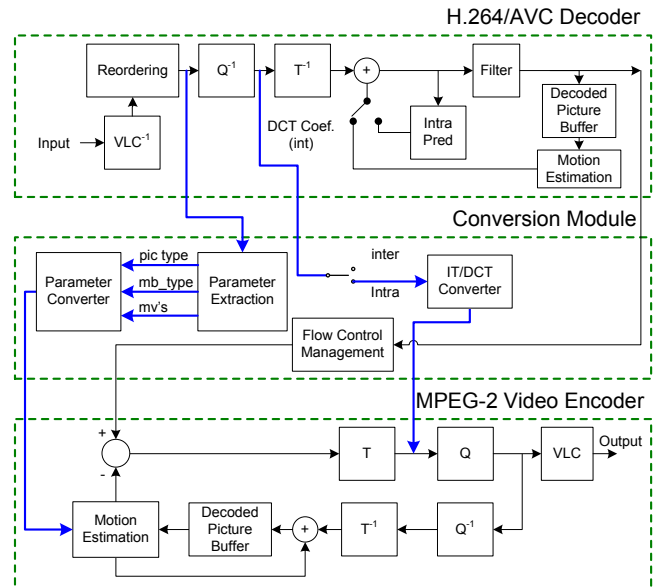
will be a significant time saving for the encoding process [7]. As both standards share the same theoretical principles for temporal prediction, the reuse of H.264/AVC previously computed motion information is still valid, as the motion properties of the encoded sequence are unchanged despite the source encoder.

In the H.264/AVC encoder, the motion estimation routines exploit the temporal correlation between frames and for each macroblock they are responsible for searching an optimum motion vector in R-D sense. Such an R-D optimisation aims to optimise the ratio between the amount of information used to encode a specific motion vector and its contribution to reduce the distortion. In order to reuse the coding mode information the *Conversion Module* needs to extract several parameters from the bitstream, such as macroblock information, picture type, motion vectors, etc. The useful parameters are converted into the MPEG-2 format, in order to avoid recalculations like full search motion estimation. Conversion of Discrete Cosine Transform coefficients to Integer Transform coefficients in intraframe coding can be also optimised [8].

2.1 H.264/AVC Decoder

In the proposed architecture, the H.264/AVC decoder is the only functional block that cannot be optimised. All decoded data from this block is essential for the system, since the input bitstream has to be fully decoded. Full decoding is also necessary because for some particular coding modes the Conversion Module just bypass all the information as no lower complexity encoding alternative can be found. For such macroblocks full decoding and encoding is used.

2.2 Conversion Module

The conversion module acts as a bridge between the H.264 /AVC decoder and the MPEG-2 encoder. It extracts several parameters from the H.264/AVC bitstream to be used at the MPEG-2 encoder. This functional block can be separated in three:

- Intraframe - consists in a IT/DCT Converter, that provides the coefficient conversion between both stan-
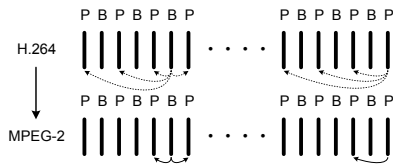
Figure 5: Motion vector scaling for B and P macroblocks.

Figure 6: Multi Reference Prediction.

dards avoiding the full decode and recode of the macroblocks.

- Interframe - responsible for the parameter extraction from the H.264/AVC bitstream and their conversion. This block extracts parameters like picture type, macroblock type, motion vectors and reference pictures. The analysis of these parameters allows to determine their use at the encoder side, namely by avoiding full search motion estimation.

- Raw Data Management - provides a flow control between the decoder and the encoder, so that a smooth frame conversion is obtained.

2.3 MPEG-2 Video Encoder

The MPEG-2 encoder was largely modified and linked with the Conversion Module in order to reuse the converted parameters originally extracted from the H.264/AVC bitstream. The encoder block that is most affected is the motion estimation where now it is possible to bypass the motion vectors for those macroblocks, where the conversion module has decided to use a direct parameter conversion. The conversion module provides a set of converted parameters previously processed, that can substitute the use of motion estimation.

The MPEG-2 motion estimation routine exploits the temporal correlation between frames, in order to find a similar area in the reference frames which minimises the residue. When the conversion module provides a motion vector candidate, the MPEG-2 motion estimation module performs a validation check for it. If this verification fails, then the macroblock is encoded using the MPEG-2 motion estimation function.

3. H.264/AVC - MPEG-2 TRANSCODING

Efficient transcoding between H.264/AVC and MPEG-2 is not a trivial operation because there are several specific coding issues that prevent a direct conversion between them, though the two standards are based on the same block-based coding approach. The transcoding techniques described in the following sections are primarily targeted at those coding modes where straightforward conversion methods do not exist. Related work described in [9] and [10] only deal with P slices whereas the proposed techniques handle both P and B slices conversion.

3.1 16×16 SKIP

The use of SKIP macroblocks is usually associated with macroblocks that maintain their characteristics and therefore the encoder can "skip" any additional information in the bitstream since it would be redundant. Note that the concept of SKIP macroblocks in the H.264/AVC and MPEG-2 standards is slightly different between them.

In H.264/AVC SKIP macroblocks in P slices correspond to either static areas or constant movement compared with the reference image. In MPEG-2 the SKIP macroblock is
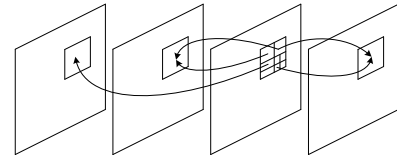
only used if the image area is static which introduces a mode conversion problem. Therefore, the macroblock conversion keeps the macroblock as SKIP for static areas and codes it as predicted for constant movement areas. In order to decode this MPEG-2 SKIP macroblock type the decoder picks the corresponding pixels from the last reference coded image without applying any motion vector neither any residue.

The mode conversion in B slices for this macroblock type is however slightly different when compared with the P slice case. The MPEG-2 standard allows to encode SKIP macroblocks in those image areas where there is either static or constant movement just like H.264/AVC. This feature allows using fast mode conversion methods very often which yields higher computational complexity savings.

Conversion of SKIP macroblocks are constrained by those MPEG-2 slice coding rules which do not allow any SKIP macroblock to be the first neither the last in the slice. Since each MPEG-2 slice cannot exceed the length of a row of macroblocks, then the number of not allowed positions is considerably high. The percentage of these type of macroblocks tends to increase as the image resolution decreases. In transcoding, When the incoming H.264/AVC SKIP macroblock is located in one of these positions, conversion is done by coding it as predicted with a null motion vector, thus avoiding the motion estimation routines.

3.2 16×16 predicted

The 16×16 predicted coding mode is a particular case because it has a unique partition with the same macroblock dimensions (16×16) in both standards. This macroblock mode is used to encode those macroblocks that failed the SKIP conversion as well as the source (16×16) ones. Although this seem to be a simple conversion, the strong differences between both standards make it a complex process.

It is known that multiple reference prediction is one of the most innovative tools introduced in H.264/AVC by allowing the use of up to 16 reference pictures in the motion estimation process. The increase in the amount of reference pictures permits the encoder to search among all reference pictures the best area that minimises the prediction error. While this introduces significant computational complexity, it results in coding efficiency gains. The MPEG-2 standard only uses one reference frame in P pictures and two in B pictures. Transcoding of macroblocks which use more than one or two reference pictures requires an additional conversion step to make it compatible with the MPEG-2 standard. When a H.264/AVC motion vector has a reference picture beyond the MPEG-2 scope, the new candidate motion vector is obtained by scaling the H.264/AVC motion vector such that the new reference picture is inside the MPEG-2 temporal limit. The scaling factor is determined by calculating the temporal distance between the current picture and the reference picture as it is illustrated in figure 5. This scale factor assumes a uniform motion of the corresponding area.

The use of unrestricted motion vectors in H.264/AVC produces non compatible MPEG-2 macroblocks since it does not allow any motion vector to point outside the frame boundaries. This H.264/AVC feature is very useful when
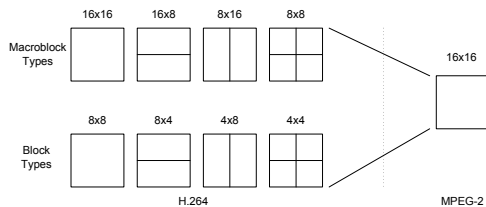
Figure 7: Macroblock Conversion.

Figure 8: Computational complexity gain.

encoding sequences with camera pannings because it allows to use image areas beyond the frame limits as reference. Macroblock conversion for this coding mode is performed with a motion vector truncation and constraining it inside the frame limits.

The pixel precision used for the motion vector represents the accuracy used at the motion estimation process. The H.264/AVC uses a quarter-pixel motion vector precision whereas MPEG-2 only uses half-pixel precision. Since the motion vector conversion involve scaling and pixel accuracy conversion, the use of rounding instead of truncation is mandatory in order to minimise the conversion error.

3.3 MB Sub-partitions

Another major feature of H.264/AVC is the possibility to break each macroblock into several 8×8 blocks . Each block can then be divided again into smaller 4×4 partitions. Such a macroblock segmentation increases the performance of the motion estimation process since each partition can achieve a better prediction than a unique macroblock. The use of several partitions reduce the prediction error. However, as the number of partitions increase, the number of motion vectors also increases. The main problem of converting a multi partitioned macroblock is that MPEG-2 does not support the same macroblock partitioning scheme and therefore it is necessary to convert the partitioned macroblocks to a single one, as illustrated in figure 7. Furthermore, H.264/AVC allows the use of one reference frame for each block as shown in figure 6, which increases the mismatch when converting it to a unique macroblock.

The candidate motion vector is selected from the several motion vectors of the macroblock partitions by computing the *Sum of Squared Differences* (SSD) and selecting the motion vector that achieves the lowest SSD value.

4. EXPERIMENTAL RESULTS

The experimental results were obtained with three different test sequences, (Mobile, Stockholm and Shields) each one with 250 frames and 720×576@25 Hz. These were selected according to their motion activity and spatial detail. The GOP size was 12 and its structure 'IBPBP' and with a 5Mbps bitrate for both the H.264/AVC and MPEG-2 streams. The H.264/AVC input stream was produced with 5 reference frames. Both encoders followed a GOP size of 12 with a 'IBPBP' structure. The simulation were performed on PC with a 3GHz processor and 1.5GB of RAM memory. These simulations pretend to evaluate the coding gain achieved by the proposed techniques in terms of computational time savings as well as the objective video quality.

As shown in figure 8 the proposed architecture achieves a computational complexity gain of up to 60 % with a reasonably low quality loss (≈0.5dB) when comparing with the full decode and reencode. It was also verified that the coding gain as well as the image quality depend on the intrinsic sequence properties achieving a better results when there is fewer spatial activity.
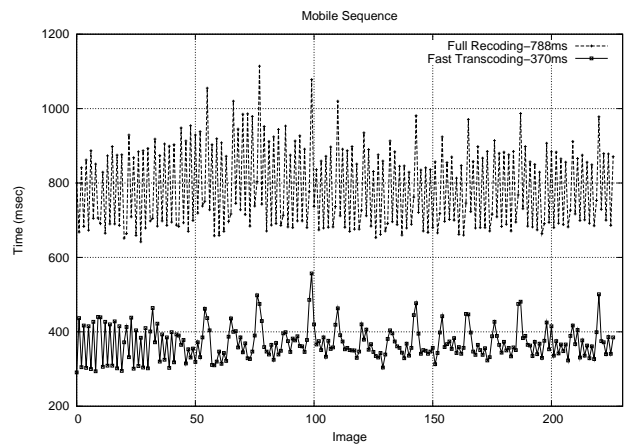
5. CONCLUSIONS

The proposed video transcoder can be used for content adaptation in wireless multimedia services where H.264/AVC compressed video is delivered to users with MPEG-2 equipment. It use as an additional home gateway functionality will provide increased compatibility in video applications. The transcoder exploits interframe coding similarities between the H.264/AVC and MPEG-2 by reusing the motion estimation embedded information in the source bitstream. The experimental results show a significant computational complexity reduction, up to 60% with an acceptable quality loss, when compared with the reference transcoder.

6. REFERENCES

[1] JVT of ISO/IEC MPEG and ITU-T VCEG, *ITU-T Rec. H.264 ISO/IEC 14 496-10 AVC)*, March 2003.

[2] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with h.264/avc: Tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, pp. 7–28, 1st Quarter 2004.

[3] ITU-T, *Recommendation H.262, Information Technology - Generic Coding of Moving Pictures and Associated Audio Information: Video*, Feb. 2000.

[4] H. Kalva, "Issues in H.264/MPEG-2 video transcoding," *Computer Science and Engineering*, 2004.

[5] http://iphome.hhi.de/suehring/tml/, *JM H.264*.

[6] http://www.mpeg.org/MSSG/, *MPEG2 v1.2*.

[7] I. Ahmad, W. Xiaohui, Y. Sun, and Y.-Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Transaction on Multimedia*, vol. 7, pp. 793–803, Oct. 2005.

[8] R. Marques, S. Faria, P. Assuncao, V. Silva, and A. Navarro, "Fast conversion of H.264/AVC integer transform coefficients into dct coefficients," *SIGMAP*, pp. 5–8, Aug. 2006.

[9] P. Kunzelmann and H. Kalva, "Reduced complexity h.264 to mpeg-2 transcoder," *Digest of Technical Papers on the International Conference on Consumer Electronics*, Jan 2007.

[10] L. Yang, X. Song, C. Hou, and J. Dai, "H.264 MPEG-2 transcoding based on personal video recorder platform," in *Proc. of the Ninth International Symposium on Consumer Electronics*, June 2005.