J Real-Time Image Proc (2016) 12:443–454
DOI 10.1007/s11554-015-0496-4

SPECIAL ISSUE PAPER

Advanced multimedia service provisioning based on efficient interoperability of adaptive streaming protocol and high efficient video coding

Jordi Mongay Batalla

Received: 22 January 2015 / Accepted: 12 March 2015 / Published online: 24 March 2015

Abstract The popularity of the new standard H.265 (High Efficiency Video Coding) depends on two factors: the rapidity of deploying clients for the most outstanding platforms and the capacity of service providers to develop efficient encoders for serving demanding multimedia services such as Live TV. Both encoders and decoders should inter-operate efficiently with the streaming protocols to limit the risks of implementation incompatibilities.This paper presents a synergy framework between High Efficiency Video Coding (HEVC) and Dynamic Adaptive Streaming over HTTP (DASH) that increases the compression capabilities of the encoder and opens new parallel encoding points for accelerating the video content coding and formatting processes. The synergy takes advantage of inter-motion prediction in HEVC for delimiting the segments of DASH protocol, which increases the motion compensation in each segment. Moreover, the impact of encoding multiple DASH representations is limited due to the parallel encoding presented in this paper.

Keywords DASH-HEVC interoperability · Video service management · Compression efficiency · Parallel encoding · Real-time HEVC encoder

1 Introduction

New services for multimedia (particularly video) transmission are constantly appearing in the network, but they revolve around two ways of serving the Multimedia content: on-demand service or in a continuous streaming (Live Television). Video on Demand is served in the following ways: Subscription Video on Demand (the client pays a subscription and may download different content, e.g., Netflix), Transactional Video on Demand (pay-per-view service), Free Video on Demand (free content). Free Video on Demand is technically much easier to be implemented, since it requires only Over-The-Top (OTT) infrastructure. Live TV services offer broadcast TV with or without interactivity from the users. A midway between Live TV and Video on Demand (VoD) is the television shifted in time (Catch-TV), which means that the television programs are recorded near the users and are accessible during a period of time (hours or days). This service is also called Push Video on Demand, but in this case, the content pushed from the servers to the video recorders is only the popular content (Push Video on Demand is used by broadcast providers for improving efficiency of the own infrastructure). A simplified version of Catch-TV is the so-called start-over TV, where the content is recorded only during the duration of the TV program transmission and a short time after it (the users may replay the content from the beginning).

Among all the afore-mentioned services, these related with Live TV are very challenging from the encoder and streamer points of view. The operations related to encoding and preparation for streaming should be performed in real-time or quasi-real-time conditions. Therefore, the performance of the encoder is a key element for the service provider's system and only high-performance encoders may be used in real scenarios. As a result of this, the time-to-market of new encoding and streaming solutions is long and involves many challenges. Many of the current commercial services use old coding technologies with encoders that have been tested and optimized for a long period of

J. Mongay Batalla (✉)
National Institute of Telecommunication & Warsaw University of Technology, Warsaw, Poland
e-mail: jordim@interfree.it
URL: http://www.nit.eu/z3

time. For example, France TV should definitively merge to MPEG-4 in 2015, so the decoders are not obliged to decode MPEG-2 anymore starting from year 2015 [1]. For a long period of time, the prevailing encoder will be MPEG-4, even when it is a fact that high efficiency video coding (HEVC) [2, 3] will be the fundamental codec of the future. At this moment, the popularity of HEVC and its use in the market depend on two factors: the rapidity of deploying clients for the most outstanding platforms and the capacity of service providers to develop efficient encoders for serving demanding multimedia services such as Live TV.

Live TV is also challenging for the streaming protocol solutions, especially in adaptive streaming, since the service provider needs to prepare different contents to be streamed in real-time. TCP-based solutions (and particularly HTTP-based) are competing for the market of TV solutions transmitted through the Internet. Over the last years, a large number of solutions for adaptive streaming have been proposed [4]. Among them, the new standard Dynamic Adaptive Streaming over HTTP (DASH) [5] is called to dominate the market due to its simplicity and flexibility. Moreover, all of the most important Multimedia software producers (Microsoft, Adobe, Samsung, Sony, etc.) have joined DASH and announced the intention of deploying software for any device using any Operating System. In fact, many DASH clients have been developed and are currently in use, especially in OTT VoD services and in hybrid television market (HbbTV) [6]. The DASH Industry Forum comprises the most outstanding network/service operators offering Live TV, therefore, it is relatively obvious that in a near future DASH will gain a significant part of the market of video streaming on the Internet.

DASH is a stream-switching adaptive protocol that is based on the download of portions of video called segments or chunks over HTTP-connections. It requires the video to be fragmentized and encoded in different bitrates (called representations). Such representations are re-called from the client to adapt the video streaming bit rate to the network situation. DASH is codec-agnostic, i.e., DASH supports any audio and video codec. Codec-agnosticism reinforces the flexibility of DASH solution; however, at the same time it represents a new challenge for implementation of DASH standard-specific players (in fact, excessive flexibility raises the risk of incompatibility of different deployments [7]). In other words, even when the standard is independent of the supported codec, the deployment is not independent anymore. For example, the encoder should be aware of the segment encapsulation needed for streaming. Therefore, DASH Industry Forum [8] has developed interoperability points parallel to the DASH standard, which are the way to take the general standard towards a defined implementation. The interoperability points are a guideline for implementers/developers to deploy concrete solutions for DASH-based distribution by defining, for example, how the implementation should take into account the given codec, how to insert advertisement, how to describe advertisement in the Media description files and which kind of Digital Rights Management (DRM) solution is supported, in addition to media timeline events support, blackout signalling, content asset identifiers and others. Thanks to interoperability, the DASH-standard has strengthened its position and made smoother deployment roadmap feasible.

After the success of DASH-AVC/264 implementation guidelines (interoperability point between DASH and H.264/AVC codec [9, 10]), DASH Industry Forum is currently working on the document (commitment from vendors and service providers) for specific implementation of DASH together with HEVC codec. It should be a powerful set of DASH characteristics for fast implementation in HEVC encoders. Moreover, the conformance and partially performance tests will be specified together with reference testing software. The final document has not been delivered yet.

This paper discusses the interoperability of DASH and HEVC and proposes some synergies between these two standards which may increase the compression and reduce the service composition (HEVC coding + DASH formatting) time. We do not analyse the HEVC coding process itself but from the point of view of the streaming protocol and we propose a synergy framework that reduces the bitstream data (compared to system without synergy) and propose points of parallel encoding (in addition to the existing ones inside of the HEVC encoder) for accelerating the encoding process. The proposed parallel encoding points come from the inter-operability between HEVC and DASH. The details about this synergy are presented in Sect. 3.

Before, Sect. 2 presents a scenario for personalised content production to be used in both Video-on-Demand and Live TV services. This scenario is an example of the necessary inter-operability of codec and streaming protocol. By the term "personalised content production" we refer to the preparation of the content (to be streamed) in the moment when the user requires the service. We are aware that the proposed scenario (containing service manager and service composition) is not real since the HEVC technology is distant of deploying real-time encoders. Therefore, even if we implemented the proposed solution, it will not be applicable into the real network since the service management presented here claims high-capacity HEVC encoder.

Section 4 faces up the validity of the DASH-HEVC synergies by testing the implemented solution. The tests aim to demonstrate the increase of video data compression

(compared to system without synergy) without losing quality. At last, Sect. 5 summarizes the contribution of the paper and explains the future work.

2 Scenario of real-time service provisioning

Last research on Content Aware Networks [11] has opened new scenarios for Multimedia delivery within the Internet. Such scenarios are very challenging from the point of view of the service management.

The Service Provider is in charge of managing the life-cycle of the media services including service planning provisioning, publishing, delivery and deletion. Service planning is the process in/which the Service Provider decides which kind of service (Live TV, Catch-TV, Video on Demand) should be accessible by the users for given content. Service provisioning consists of preparing the content (content composition) for live or on-demand transmission. The content is then stored in the content servers following the business plan (Content Delivery Network, own content servers, etc.). Publishing is the popularisation of the content to be accessible by the users. At last, the content should be delivered immediately after the user request. In Live TV, the delivery of the content is performed regardless of the user requests. The Service Provider is also responsible for deleting the content or cancelling the access to the content, when needed. The Service Provider manages the whole process by means of the service manager.

In new architectures of service manager, the service may be provisioned on-demand of user request, what opens new business models (e.g., personalized advertisement, terminal-specific streaming), which are very interesting from the point of view of monetization. In this case, the service provisioning should begin when the end user's request arrives to the system and should make use of the available information about the user. The service manager is in charge of controlling the whole provisioning process: from the arrival of the user's request until the content storage and the final notification of readiness to the end user.

We propose an integrated provisioning system capable of preparing the content for streaming based on the user request, when this request arrives to the service manager (see Fig. 1). The proposed service provisioning manager composes content in accordance with the following standards: H.265 (commonly named HEVC) and DASH. The advantages of the proposed system are countless (personalised high-quality Multimedia services with adaptation capabilities). The inconvenient is the necessity of real-time HEVC encoders, which is not real at this moment. Anyway, the research on HEVC is following really fast in this direction.

When the user requests a new content, the service provider manager initiates the discovery phase and prepares the content for the user by selecting the appropriate codec and streaming features (e.g., potential download rates or representations) to the user's context and by adding personalised advertisements (ads). Personalization is based on the information disclosed by the end user about one's her/his profile, preferences and context. The end user maintains updated database of the current context of the terminal and sends this information during the content request.

A business key functionality in the service life-cycle is advertisement (ad) insertion and, inside this, on-the-fly personalized advertisement insertion is one of the most

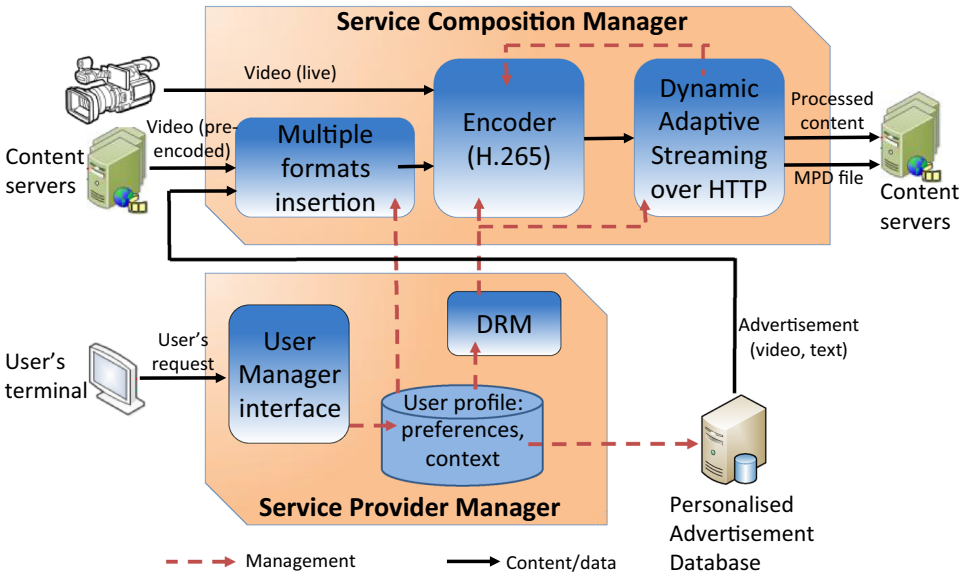

Fig. 1 Service provisioning manager based on user requests

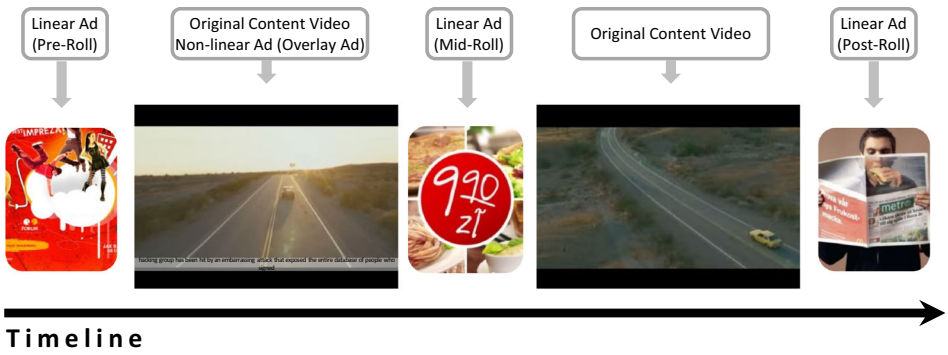

Fig. 2 Types of advertisement insertion

promising assets in Multimedia streaming market. The proposed service manager uses User awareness (user context, profile and preferences) to optimize personalized ads insertion. Up till now, user profile has only been considered in ads insertion [12, 13] (note that the authors of [12] use the term User context for referring to the user profile). Our solution of service manager also includes the user context for increasing the quality of the user's viewing. For example, if the user has limited bandwidth, then linear ads are preferable to non-linear ones (see Fig. 2).

Personalized ads insertion is performed when a user content request arrives to the Service Manager (see Fig. 1). The SM performs the first selection of advertising content location based on the user context. Afterwards, the SM selects the appropriate ad content from the third party ads server to be inserted into the requested original content according to the user profile (language, personal data) and user preferences (advertising preferences).

The system is able to render Linear and Non-Linear advertising (see Fig. 2). Linear advertising refers to pieces of video embedded into the original content whereas Non-Linear advertising refers to video or text overlapped with the original content (e.g., Logo). The Linear advertisement is composed of a number of segments (chunks) that are independent to the content segments and are mixed at the level of the streaming protocol (in our case: DASH). From the deployment point of view, due to segment division of DASH, Linear advertisement must not be encoded together with the movie, but it is an added piece of video and will be encoded and prepared for streaming in the same way as any other video. The Non-Linear advertisement is added to the video content by multiple formats insertion module situated in the service composition manager (see Fig. 1). Since the segments are generally short, ads insertion + encoding + streaming protocol formatting should be performed in a very short time, which would create on-the-fly perception from the user point of view (one must not wait a long time until receiving the first segment of content). Transmission formats such as DASH require the inclusion of ad url into the manifest file in a dynamic mode [14]. Therefore, the user profile module sends information about ad insertion to the DASH library into the service composition manager.

The description of advertising makes use of Video Ad Serving Template (VAST) [15], which describes expected video player behavior when executing VAST formatted advertising. VAST is an XML standard schema specification that has been designed for serving ads to digital video players of the End User Terminal [15]. In our case, we decided to perform ads insertion in the service manager because of two main reasons: (1) generally, the End user Terminal does not have enough hardware resources to perform ads insertion during the play of the content, and (2) insertion of advertisement at the service provider's side respects integrality of the content, which increases the value of advertisement inside of the content.

The next section shows details of the service composition manager (Fig. 1) and, specially, of the interoperation of HEVC encoder and DASH streamer.

3 Synergy framework between HEVC and DASH

The service composition manager presented in this paper integrates HEVC encoding and DASH format embedment of content. The content is then stored in the content servers and may be immediately streamed to the end users "on demand".

The interworking of HEVC encoder and DASH format creation modules makes enhanced compression feasible, basically due to the coordination of HEVC inter-picture (motion) prediction and compensation with DASH formatting (parameters). In previous papers, some authors presented the interwork of decoders and encoders (e.g., AVC decoder and HEVC encoder) for reducing the complexity of HEVC encoder by taking advantage of the information provided by the AVC decoder as, for example, motion information [16]. Anyway, at the best of our knowledge, no papers proposed interworking between codecs and streamers.

Moreover, we propose to use the cooperation framework between HEVC and DASH for finding new parallel encoding points for accelerating the encoding process for DASH-HEVC content. Specifically, we propose to separate the encoding work for different DASH segments and for different DASH representations.

In the following sub-section, we present the constraints of DASH standard for the final bitstream that are directly related to the video coding. In addition, we present HEVC characteristics that are considered in the synergy framework, especially the features of motion estimation and compensation, which are directly related to DASH-HEVC interoperability. Note that we do not aim to present all the details regarding DASH and HEVC, but only those, which are related to the modifications provided in our service composition manager. Sub-sect. 3.2 presents the details of the service composition manager and, in Sub-sect. 3.3, we propose the solution for the DASH-HEVC enhanced encoder working jointly with the service manager presented in previous section. At last, Sect. 3.4 presents the parallel encoding points risen from the DASH-HEVC interoperability.

3.1 DASH and HEVC features used in our solution

DASH constraints for coded bitstream. The DASH standard for on-demand and live services is defined by the use of ISO base media file format (ISO BMFF) as defined in ISO/IEC 23009-1 [14]. In practice, the ISO BMFF requires the coded bitstream to be one segment with a number of sub-segments for on-demand service or a set of segments for the live service. Each segment or sub-segment should start with a Stream Access Point for independent decoding. In addition to this, segments and sub-segments should avoid gaps and overlaps of video between them, so the switching between segments or sub-segments is smooth (also in the case of representation switching). Each fragmented video should be used for encapsulation of media data. In this way, the segments (or sub-segments) are standalone and may be decoded independently of other segments.

The key functionality of DASH for inter-operating with fragmented video is the timeline. All the fragments of the same video share the same timeline. The way as the video fragments are re-called into the timeline is organized by the addressing scheme. There are four potential addressing schemes in DASH : (1) templates with number-based addressing, (2) templates with time-based addressing, (3) templates with byte-based addressing, and (4) templates with playlist-based addressing, however, the last one is not valid for interworking with HEVC video in on-demand or live services due to the fact that playlist-based addressing is reserved for the situation of several clients with access to distributed content in a dynamic way (dynamic Media Presentation Description), which is not the case with the specified services.

Even when the DASH-HEVC interoperability points have not been defined yet, the most of DASH constraints for the coded bitstream are the same as in the case of interoperability of DASH and H.264 since those constraints depend on the DASH standard requirements. One of the requirements is the equal duration of successive segments (or sub-segments), which should be the same for given content and given service with a tolerance of 50 % of the maximum duration defined in the Media Presentation Description (MPD). The MPD is the file defining the segments, representations, URL addresses and all the information related to the media streaming. Generally, changes in duration are permitted for advertisement purposes, however, in our solution we used this tolerance for increasing the performance of the encoder by means of adapting the segment duration to the motion variability of the movie, as it will be explained below. In our service composition manager, the linear advertisements are encoded in separated segments and, also in this case, they can make use of the segment duration tolerance.

Other restrictions of DASH standard related to the kind of service offered by the service provider are the conformance of the MPD type (static or dynamic) to the service. Static MPD means that the segments are accessible from the moment when the content is published, whereas dynamic MPD means that the segments are accessible at the moment of publishing the content plus a period of time that will increase for successive segments. On-demand services should be managed by static MPD, whereas live services can be managed by both static and dynamic MPD. Often in live services, the MPD must be updated.

HEVC motion prediction and compensation. Since successive frames in any movie contain similar information [17], one of the ways to comprise the information is to eliminate the temporal redundancy between frames. The first picture of any access point into a video movie is encoded without any compression of time information. These pictures have only spatial intra-picture compression; therefore, the picture can be decoded with the streamed information about itself without the necessity of comparing with other pictures.

The rest of the pictures are divided into blocks and the motion compensated prediction [18] estimates the motion of different blocks by comparing them with the reference pictures. Once the best correspondences are found, then the differences (original minus predicted signal) is encoded and sent along with the video signal. In this way, the decoder owns all the information necessary to decode the image without losing information. In addition, the motion vector (relation of the position of the block with regard to

the reference picture) is also encoded and sent with the signal.

The motion vector in HEVC may apply fractional accuracy, which reduces the residual error by avoiding additional information in the encoded signal. HEVC allows vectors moving a quarter of a pixel in successive pictures (H.264 allows movements down to 0.125 pixels of chroma vector). The key issue of fractional values of the motion vector is that each block of the reference picture (matching block) should be interpolated following the fractional accuracy. The main advances of HEVC compared to previous H.264 standard are related to the interpolation filter. The interpolation filters used in HEVC are finite impulse response (FIR) filters and the number of parameters in the filter are increased (8 coefficients for 0.5-pixel movements and 7 for 0.25-pixel movements) to improve the performance in high frequencies.

The gain of HEVC efficiency (compression) is around 4 % in luma components and around 11 % in chroma components [19], but the difference in efficiency gain is due to the fact that chroma components in H.264 had a maximum accuracy of one-eighth of a pixel, which proved that it does not offer a good trade-off between accuracy and signal overhead.

The higher efficiency in HEVC is obtained due to the high accuracy maintained during the interpolation operations. In fact, HEVC interpolation filter does not reduce the accuracy during intermediate operations, as H.264 did. The maintenance of accuracy is obtained since the filtering process is not performed in cascade, but it is done in parallel by conserving two-dimensional (one for each direction) values of filtering operations. This means that the rounding operations are performed only when the bi-directional values are summed up (at the end of the filtering process) and not during bi-directional operations.

The complexity of HEVC encoders is particularly caused by the motion compensation algorithms, which may consume half of the code complexity (both number of code lines and computational time of the operations) [20], due to the extended number of prediction modes and blocks that must be evaluated for rate-distortion optimization [21]. Therefore, in the next sub-sections we propose to take benefit of the characteristics of DASH to reduce the complexity of the whole interoperable DASH-HEVC encoder.

3.2 Service composition manager

The service composition manager proposed in this paper integrates operations of DASH encapsulation with HEVC encoder. In this way, the complete solution takes advantage of the information that the HEVC encoder owns about the video pictures to appropriately perform the division in segments within the DASH formatter. Moreover, the information of the representation rates necessary at the DASH level is used for adjusting parameters within the HEVC encoder.

The video (live or not) arrives to the service composition manager (see Fig. 1) to be transcoded/encoded and formatted in DASH segments for different representations. The multiple formats insertion module is in charge of selecting and ordering the video to be encoded following the instructions of the User profile into the service manager. The multiple formats insertion module decides in which moments the advertisement should be encoded and inserted in the final encoded video. Note that the advertisement is inserted in separated DASH segments, so the encoding and formatting of ad video is independent of the rest of video content. Nonetheless, multiple formats insertion should decide (instructed by the service manager) when to encode the advertisement content to generate continuous (from the end user's point of view) content (video + ads), which may be served on live to the end users.

In the case that the original video (arriving to the service composition manager) is encoded in any other codec and must be transcoded to HEVC, some of the parameters of the original codec (e.g., motion estimation parameters) could be used for accelerating the HEVC encoding. Several papers proposed similar approaches, especially for AVC-to-HEVC transcoding [22, 23]. This functionality has not been yet implemented in our solution.

Since ad insertion and transcoding are performed on a per-request basis, high performance capabilities are required to the multiple formats creation module. The proposed implementation performs dynamic content provisioning by ingesting new content items and new formats of existing content. The tasks of the module are queued and reported by a task scheduler, which manages the operations into the service composition manager and is in communication with the service manager. The task scheduler controls the edit decision list, which has been developed in the multiple formats creation module to precise internal parameters of tasks for input/output, encoding and Digital Rights Management encryption. This schema allows building complex processing jobs by combination of basic tasks. The <destination> element in I/O section fixes delivery servers and repositories where to copy the output content.

Transcoding and ad insertion are created in the multiple formats insertion in two manners: watch folder mechanism where processing tasks are created and started automatically whenever video files are arriving in pre-defined folders, and by the use of Application Programming Interface, API (RESTful and SOAP protocols). API enables reusability and agile development on multiple client modules. The multiple formats insertion module triggers the coding operations into the HEVC encoder, as it will be explained in the next sub-section.

3.3 DASH-HEVC encoder

When a new video arrives to the HEVC encoder, it is cut by the frame cutter in a number of sub-frames with established duration. The sub-frames are the result of cutting one frame in fragments with the same or similar duration. We call a frame as a set of pictures containing the same motion vectors. The duration of all the sub-frames obtained from the fragmentation of one frame is the same and it is calculated based on the maximum DASH segment duration (called *max_seg_duration*) defined into the MPD file. Concretely, the first operation performed on the video pictures is a designed motion estimation module that investigates the points in the video timeline where the different frames start. Once a frame (named *frame_f*) is discovered, the frame cutter algorithm calculates the duration of the frame, duration (*frame_f*), and applies the algorithm presented in Listing 1. Briefly, the frame cutter algorithm cuts each frame in a number of sub-frames with duration equal to *max_seg_duration*, except the last sub-frame that must have a long major or equal to the 50 % of *max_seg_duration*. If it is not possible, then two last sub-frames will have a duration between $0.5 \times max\_seg\_duration$ and $1.0 \times max\_seg\_duration$. In line 3 of Listing 1, the algorithm finds an error for frames that are shorter than the 50 % of *max_seg_duration*. In this case, the cutter must put together at least two different frames. The proposed algorithm is applied to each video encoded in HEVC, independently of the nature of the video: low or fast motion, the quality of the video: HD or SD and the purpose of the video: entertainment, advertisement, etc.

The result of the frame cutter algorithm is a number of video fragments with similar duration (the duration of the fragment follows DASH standard) and where the image is similar (same motion vector) in the whole fragment. Thanks to that, the HEVC encoder may comprise each segment with high efficiency, as it will be shown in the test results (Sect. 4).

The block scheme of the DASH-HEVC encoder is presented in Fig. 3. All the modules depicted in red are directed to the proposed DASH-HEVC interwork. Instead, the yellow modules are the typical modules of an HEVC encoder. In our implementation, we used the HEVC code developed by Franhofer Institute [24, 25]. We used the version 8.0 of the publicly available implementation, which was the latest available and stable version when we started our implementation [26]. We did not provide significant modifications to the modules, but we introduced an interface between the multiple formats insertion module and some of the modules of the DASH-HEVC encoder for making the parallel coding (presented in the next subsection) feasible. The communication with multiple formats insertion module is presented by red lines and parameters in Fig. 3.

At last, the blue modules of Fig. 3 are directed to format the segments introducing DASH headers and, on the other hand, to prepare the MPD file for the encoded content. When the process of HEVC encoding is finished (bitstream of HEVC video fragmented and formatted in DASH segments), the segments of video and the MPD file may be stored in content server/s, where they will be streamed from (see Fig. 1). From the implementation point of view, we deployed two protocols for the communication with the content servers, which are used as source (for storage): usual FTP and NFS upload.

3.4 Parallel encoding points

One of the most important features for obtaining fast encoding is the capacity (of the encoders) of running parallel encoding. This is even more important in the case of multiple representations DASH content, where the content is encoded in different bitrates for media adaptation purposes. Therefore, in our solution we performed three levels of parallel encoding, which can be active together or not. The levels of parallel encoding are managed by the multiple formats insertion module, which decides on their

```
   ***FRAME CUTTER ALGORITHM***
1  %input: frame_f, max_seg_duration
2  %output: {sub-frame_f[i], sub-frame_f_dur[i]}
3  If duration(frame_f)< max_seg_duration -0.5×max_seg_duration then error_too_short
4  If remainder(duration(frame_f)/max_seg_duration)>0.5×max_seg_duration then{
5   for i=1 to N-1: sub-frame_f[i]= cut(frame_f/max_seg_duration; i);
6   sub-frame_f[N]=remainder(frame_f/max_seg_duration) }
7  If remainder(duration(frame_f)/max_seg_duration)<0.5×max_seg_duration then{
8   for i=1 to N-1: sub-frame_f[i]= cut(frame_f/max_seg_duration; i);
9   sub-frame_f[N]= cut(sub-frame_f[N-1]/2; 2)+ remainder(frame_f/max_seg_duration);
10  sub-frame_f[N-1]= cut(sub-frame_f[N-1]/2; 1) }
```

Listing 1: Frame cutter algorithm (pseudo-code)

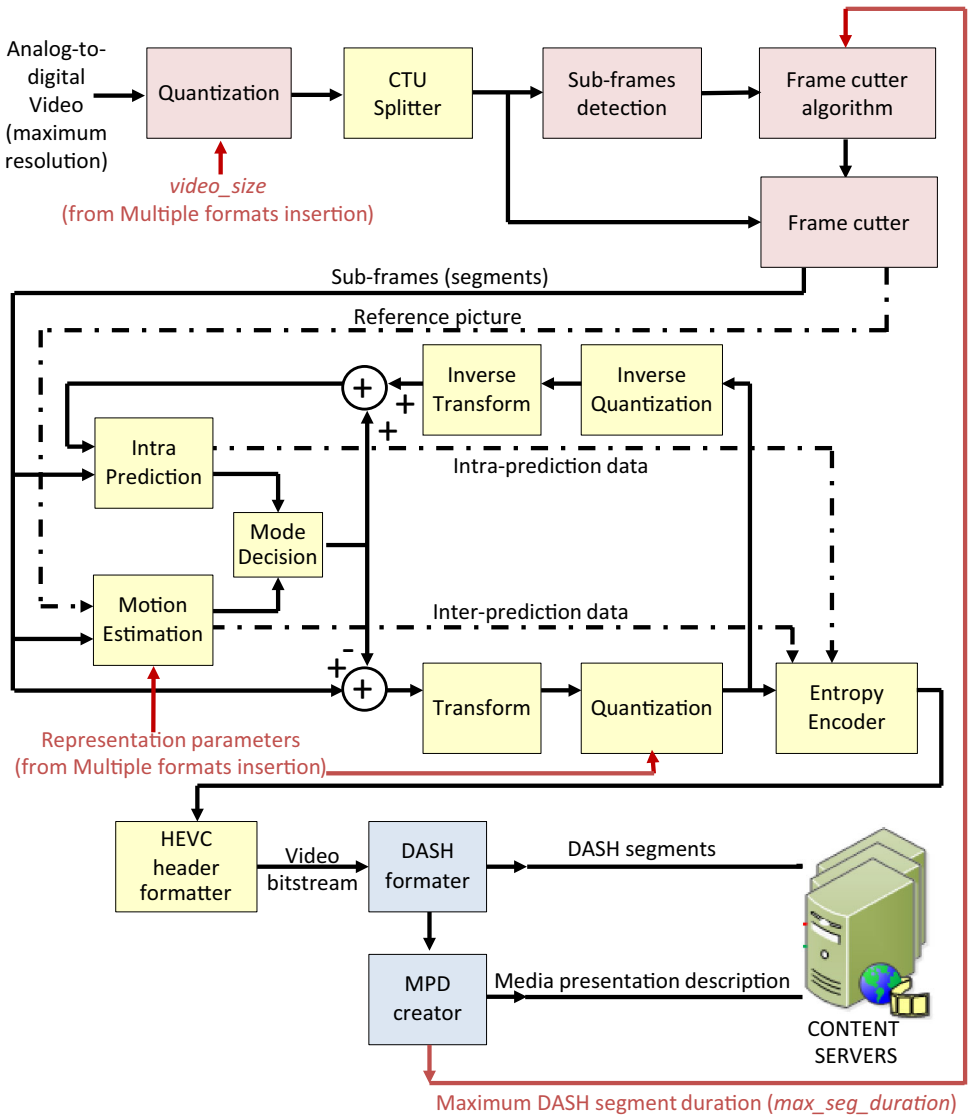

Fig. 3 Block diagram of DASH-HEVC encoder (in *red*, all the modifications with regard to reference HEVC encoder)

activation for each video arrived to the service composition manager. The multiple formats insertion module is partially instructed by the service manager about the parallel work to be provided. The information passed by the service manager refers to the necessity of encoding different video resolutions (which means multiple representations encoding) and the possibility of the user (and, eventually, of the network) of adapting the content to the network dynamicity (which also requires multiple representations encoding). This information is related to the user profile (e.g., bandwidth of the access to the network), to the user's terminal capabilities (e.g., screen size and resolution) and to the user's preferences (e.g., fast video download with low quality instead of heavy download of better video).

The three levels of parallel encoding introduced in our service composition manager are: (1) parallel video resolution encoding, (2) parallel segments encoding and (3) parallel representations encoding.

The *parallel video resolution encoding* is run when the video is quantized at the beginning of the encoding process (before of the HEVC encoding). This process corresponds to the quantization during the analog-to-digital conversion if the signal was originally an analog video signal. Also, an initial quantization process is necessary in the case when the digital raw video signal contains too much information to be processed; in this case, the first step could be the reduction of the resolution for reducing overhead in encoding operations. In these cases, the video resolution after quantization depends uniquely on the resolution requested by the user and the network (this information arrives from the service manager). Instead, if the original signal was an encoded (other codec) video, it must be firstly decoded for encoding in HEVC format and, then, the maximum resolution of the total transcoding operation depends also on the resolution of the arrived video.

The parameter *video_size* indicates the resolution of the pictures and is passed from the multiple formats insertion module to the quantization module (see Fig. 3). The quantization module decides the number of pixels and bits per pixel during the video quantization based on the value of *video_size*. Typical values of *video_size* are, for example, 640 ×360 for Standard Definition, and 1280 × 720 or 1920 × 1080 for High Definition.

The *parallel segments encoding* is executed after the segment division performed by the frame cutter and consists of encoding in parallel the different segments of one quantized video. Parallel segments encoding is especially useful for achieving high performance of coding in VoD services. Instead, Live TV service makes more difficult the parallel work of coding segments due to the real-time requirements of the service. An implementation key point in Live TV services is the length of segments. Long segments introduce delay in the encoder but allow for higher coding efficiency. Short segments allow for faster transmission to the end users but, in this case, the cost is a poorer coding process since motion compensation in short segments gets lower compression level.

At last, *parallel representations encoding* is obtained by encoding the same segment (of the same quantized video) with different encoding parameters in parallel processors. The parallel encoding output is the segment encoded with slightly different quality which may be streamed with different bitrates by the DASH server. Moreover, the DASH client (at the user's terminal) will be able to adapt the bitrate of the transmission (by selecting one of the video segments encoded with different bitrate) to the network conditions during the video streaming.

In our implementation (see Fig. 3), the parameters adjusted for parallel representation encoding are related to the motion estimation and to the quantization of the residuals. The motion estimation and compensation module sets the parameters of the interpolation filter (finite impulse response) for adjusting the image quality (and representation bitrate). The quantization level of the residual signal definitively influences on the final quality of the image and is crucial for the resulting bitrate.

The values of the parameters inserted in the modules (motion estimation and quantization of residual) depend on the number of representations necessary for one resolution (e.g., three different representation bitrates for 1920×1080 resolution), which are managed by the service manager following the characteristics of the user context (network access, user's terminal) and the state of the network. The number of necessary representations is communicated to the multiple formats insertion module and this is in charge of instructing the motion estimator and the quantization module (Fig. 3).

Note that the purpose of parallel video resolution and parallel representations encoding is the fast encoding of different DASH representations (for different DASH profiles); nonetheless, the bitrate differences between two videos of different resolutions are higher than the bitrate differences of two videos with same resolution different encoding parameters. In other words, the influence of the raw video signal quantization on the compression is decisively higher than the influence of the other parameters inside of the HEVC encoder.

4 Test of compression efficiency

The presented modules were implemented by using the HM 8.0 reference implementation provided by [24]. We added some modules for introducing the DASH-HEVC interwork (red modules in Fig. 3). In addition, we modified some existing modules for introducing an interface with the multiple formats insertion module, which may set the coding parameters, as explained above. As a result, the final modules were fully integrated into the service composition manager. All the new modules are related to video encoding; the audio coding is not optimized and its bitrate equals 512 kbps for high quality 5.1 Audio.

The tests for video (only) aim to check whether the proposed enhancements (DASH-HEVC interwork) improve the compression efficiency in comparison to "unaware" HEVC encoder and DASH formatter. The comparison-based tests consist of encoding 2 different movies in three different ways: the first coding is only HEVC encoding and the bitstream is formatted in one unique HTTP file (no DASH segmentation). The second way is to divide the content in segments of equal duration (2 s) and, afterwards, to encode and format the DASH segments. The third coding is performed by applying the system presented in Sect. 3, i.e., the movie is first analyzed and smartly cut for having segments of different but similar duration (the segment duration was 2 s ± 50 %) containing a piece of video of the same sequence (frame). Also in this case, the segments are formatted in DASH.

The two movies were one film and one basketball match. We selected those movies to understand the behavior of the encoding techniques with films of different motion dynamicity. Each movie was 10 min long. We set different resolutions to the movies: the film was encoded with three resolutions: 1920 × 1080, 1280 × 720 and 640 × 360; whereas the sport was encoded with two different resolutions: 1920 × 1080, 1280 × 720.

The comparison metrics in the tests are: (1) the total size of final formatted files (DASH), which is proportional to the compression rate, since the quality of the encoded

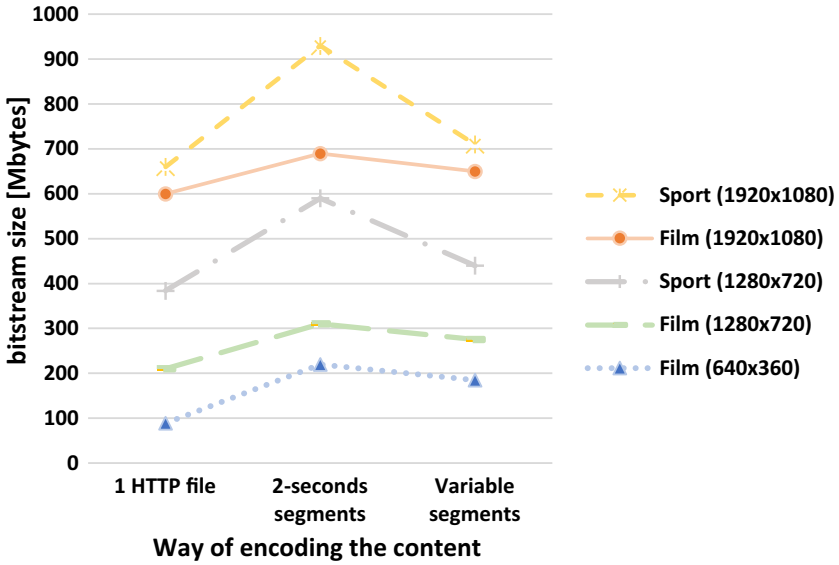

Fig. 4 Size of the whole bitstream (all the files) for the three proposed ways of encoding

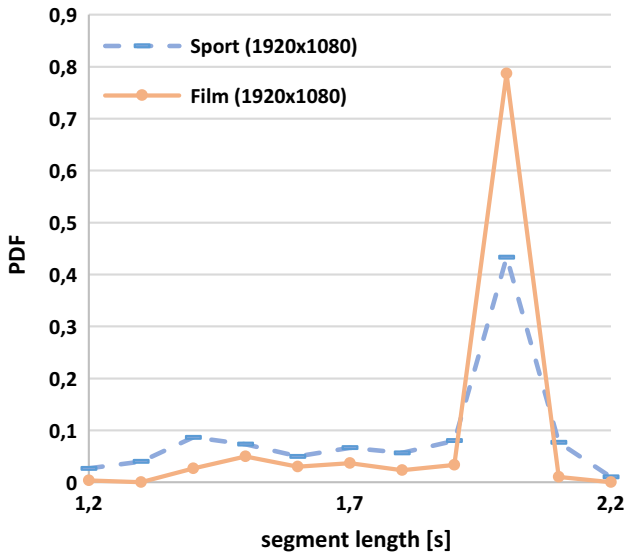

Fig. 5 Probability density function of segment length for the variable segment encoding

movies is the same regardless of the coding manner; and (2) the encoding processing time, which is a measure of the complexity of the algorithms [27]. Moreover, we analyzed the segment duration of the content encoded with variable segment duration to understand better the results.

Figure 4 presents the size of the movies in Mbytes. The values of the figure are the sizes of all the film, which in the case of movies cut in segments; it is the sum of all the segment files (the whole bitstream). In the results we can observe that coding in 1 unique HTTP file is more convenient than in segments due to the fewer overhead. In addition, the inter-prediction estimation and prediction modules contain the information of the whole content and may compress better the motion information. This is the reason by which the bitstream size gap between 1 HTTP file and the 2-seconds segments is larger for the sport movie (higher dynamicity).

The most important conclusion is that the content encoded with the proposed synergy framework reaches better compression than the content encoded without awareness between codec and streaming protocol (2-seconds segments). The reason is the more effective motion compensation since the variable segments contain pictures of the same sequence, whereas the constant segments contain pictures of different sequences.

We may observe that the bitstream size relation between 2-seconds segments and variable segments changes from the sport movie to the film movie. In fact, the relation (bitstream size for 2-seconds segments divided by bitstream size for variable segments) for the sport movie is higher than 1.3 for both the resolutions and it is lower than 1.2 for the three resolutions of the film movie. To understand these differences, we present in Fig. 5 the probability density function of the segment length of the film movie and the sport movie for the content encoded with variable segment length.

The figure shows that the most of segments in the film movie have 2 s (default value in DASH) while in the sport movie the percent of 2-seconds segment is much lower. This shows that the variability of sport movie (motion) is higher and, because of this, the algorithm selects many shorter sub-frames. In the opposite, the film movie contains longer sequences that comprise many segments and, therefore, the film movie encoded by the constant duration segments way and the variable duration segments way offers more similar values than in the case of sport movie.

At last, Table 1 presents the values of coding time for the three encoding ways. The movies are 10 min long and

Table 1 Encoding time for movies of 10 min duration

	1 HTTP file (min)	2-Seconds segments (min)	Variable segments (min)
Sport (1920 × 1080)	810	660	700
Film (1920 × 1080)	690	600	630
Sport (1280 × 720)	640	470	490
Film (1280 × 720)	590	430	480
Film (640 × 360)	340	210	230

the shown times refer to the whole movie. Let us remark that the test implementation run in HP ProLiant DL360G6 server.

The 2-seconds segments are encoded very fast since the number of operations are much fewer in this case. The variable segments has the same operations than 2-seconds segments plus the first motion estimation operation on the whole content. This is the reason of the longer encoding time. Anyway, both of the solutions are similar in comparison to the 1 HTTP file encoding, which is the most complex since the number of parallel operations is very low.

In conclusion, the proposed synergy framework improves the compression rate of the encoded video compared to the DASH formatter and HEVC encoder running independently. The encoding of the movies in 1 unique bitstream (1 HTTP file) offers better compression at encoding time's expenses.

5 Conclusions and future work

In this paper, we argue over the interoperability of codecs and streaming protocols and proposes a system for efficient interaction between the two most outstanding standards of codec and streaming protocol: HEVC and DASH, respectively. The election of these standards is due to their increasing popularity.

We proposed a system for cooperation of DASH and HEVC for increasing the compression efficiency in the encoder. The results suggested a decrease of 20–35 % in the size of the encoded content. This means a similar reduction of the bandwidth necessary for transmitting the content in relation to the transmission of HEVC content streamed into DASH segments, when DASH and HEVC do not cooperate.

Moreover, the paper presents a scenario for Video on Demand and Live TV services which claim for real-time HEVC encoder. The scenario assumes that the video content is prepared when the user's request arrives to the system. Then, the system prepares the content for streaming to the user considering the user context (user's terminal, access to the network, etc.), the user profile and the user preferences (e.g., for personalized advertisement). The potential market revenues of such a scenario are countless, which puts pressure on finding efficient solutions for real-time or quasi-real-time encoders.

Future work will be directed to the improvement of the encoder by taking advantage of transcoding facilities (original codec parameters) and by exploiting other codec parameters to obtain diversity in representations bitrate.

Moreover, the encoder development will be adapted (if needed) to the final interoperability document between DASH and HEVC that is arriving in the next months.

Acknowledgments This work is supported by the European research project DISEDAN (http://wp2.tele.pw.edu.pl/disedan/) under the CHIST-ERA framework program. The author would like to thank Marta Mongay Miland (KC) for her support during writing the paper.

References

1. Boyon, M., Rapport sur l'avenir de la Television numerique terrestre (2011). Retrieved from http://guildedesscenaristes.org/uploads/ressbao/rapports-etudes/Rapport%20Boyon%20avenir%20TNT.pdf
2. Sullivan, G.J., Ohm, J.-R., Han, W.-J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuit. Syst. Video Technol. **22**(12), 1648–1667 (2012)
3. Bross, B., Han, W.-J., Ohm, J.-R., Sullivan, G.J., Wiegand, T.: High efficiency video coding (HEVC) text specification Draft 9, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-K1003. Shanghai, China (2012)
4. Wiśniewski, P., Bęben, A., Mongay Batalla, J., Krawiec, P.: "On delimiting video rebuffering for stream switching adaptive applications". IEEE International Conference on Communications ICC. London (2015)
5. ISO/IEC 23009-2: Information technology—Dynamic adaptive streaming over HTTP (DASH), (2013)
6. HbbTV Association: HbbTV® Specification Version 1.5 (2012)
7. "Guidelines for Implementation: DASH-IF Interoperability Points". Version 2.7 (2013)
8. DASH Industry Forum: http://www.dashif.org/
9. ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), version 16, (2012)
10. Psannis, K., Ishibashi, Y.: Enhanced H.264/AVC stream switching over varying bandwidth networks. IEICE ELEX J. **5**(19), 827–832 (2008)
11. Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D.: A survey of information-centric networking. Commun. Mag. IEEE **50**(7), 26–36 (2012)
12. Song, S., Moustafa, H., Afifi, H.: "Context-Aware IPTV"; 12th IFIP/IEEE International Conf. on Management of Multimedia and Mobile Networks and Services: wired-wireless multimedia networks and services management, pp. 189–194 (2009)

13. Hua, X.-S., Mei, T., Hanjalic, A. (eds.) Online multimedia advertising: techniques and technologies. IGI Global (2011)
14. ISO/IEC 23009-1:2014, Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation description and segment formats (2014)
15. VAST (vast video ad serving template) specification—http://www.iab.net/media/file/VASTv3.0.pdf
16. Chen, Z., Tseng, C., Chang, P.:"Fast inter prediction for H.264 to HEVC Transcoding", 3rd International Conference on Multimedia Technology (ICMT) (2013)
17. Psannis, K.: Efficient redundant frames encoding algorithm for streaming video over error prone wireless channels. IEICE ELEX J. **6**(21), 1497–1502 (2009)
18. Flierl, M., Wiegand, T., Girod, B.: Rate-constrained multi-hypothesis prediction for motion-compensated video compression. IEEE Trans. Circuits Syst. Video Technol. **12**(11), 957–969 (2002)
19. Ulgur, K. et al.: Motion Compensated prediction and interpolation filter design in H.265/HEVC. IEEE J. Sel. Topics Signal Process. **7**(6), 946–956 (2013)
20. Davis P., Marikkamnan, S.: "Implementation of motion estimation algorithm for h.265/HEVC", International Conference on Signal Processing, Embedded System and Communication Technologies and their applications for sustainable and renewable energy ICSECSRE. Tamil Nadu (2014)
21. Vanne, J., Viitanen, M., Haimalainen, T.D.: Efficient mode decision schemes for HEVC Inter prediction. IEEE Trans. Circuit. Syst. Video Technol. **24**(9), 1579–1593 (2014)
22. Peixoto, E., Izquierdo, E.: A complexity-scalable transcoder from H.264/AVC to the new HEVC Codec. Image Processing (ICIP), pp. 737–740 (2012)
23. Zhang, D., Li, B., Xu, J., Li, H.: Fast transcoding from H.264/AVC To high efficiency video coding. Multimedia and Expo (ICME), pp. 651–656 (2012)
24. Joint collaborative team on video coding reference software, ver. HM 8.0. http://www.hevc.hhi.franhofer.de
25. Li, B., Sullivan, G.J., Xu, J.: Comparison of compression performance of HEVC Draft 9 with AVC high profile and performance of HM9.0 with temporal scalability characteristics. JCTVC-L0322, 12th JCT-VC meeting, Geneva (2013)
26. Bossen, F.: Common HM test conditions and software reference configurations. Document of Joint Collaborative Team on Video Coding, JCTVC-H1100, (2012)
27. Correa, G., Assuncao, P., Agostini, L., da Silva Cruz, L.A.: Performance and computational complexity assessment of high-efficiency video encoders. Circuit. Syst. Video Technol. IEEE Trans. **22**(12), 1899–1909 (2012)

Jordi Mongay Batalla was born in Barcelona (Spain) in 1975. He received M.Sc. degree from Universitat Politecnica de Valencia (2000) and Ph.D. degree from Warsaw University of Technology (2009), where he still works as Assistant Professor. In the past he worked in Telcordia Poland (Ericsson R&D) and he is still with National Institute of Telecommunications, where, from 2010, he is Head of Internet Architectures and Applications Department. Dr. Jordi Mongay Batalla took part (coordination and/or participation) in 12 international ICT projects, five of them inside the EU Framework Programmes. His research interest focuses mainly on Quality of Service (Diffserv, NGN) in both IPv4 and IPv6 infrastructures, Future Internet architectures (Content Aware Networks, Information Centric Networks) as well as applications for Future Internet (Internet of Things, Smart Cities, IPTV). He is author or co-author of morethan 100 papers published in books, international and national journals and conference proceedings.