On the use of hierarchical prediction structures for efficient summary generation of H.264/AVC bitstreams[1]

Luis Herranz

Video Processing and Understanding Lab, Escuela Politécnica Superior
Universidad Autónoma de Madrid, Madrid, Spain
luis.herranz@uam.es

José M. Martínez

Video Processing and Understanding Lab, Escuela Politécnica Superior
Universidad Autónoma de Madrid, Madrid, Spain
josem.martinez@uam.es

[1] Work partially supported by the Ministerio de Ciencia y Tecnología of the Spanish Government under project TIN2007-65400 (SemanticVideo) and by the Comunidad de Madrid under project S-0505-TIC-0223 (ProMultiDis).
The final publication is available at Elsevier via http://dx.doi.org/10.1016/j.image.2009.06.003

Abstract Video summarization refers to an important set of abstraction techniques aimed to provide a compact representation of the video essential to effectively browse and retrieve video content from multimedia repositories. Most of these video summarization techniques, such as image storyboards, video skims and fast previews, are based on selecting some frames or segments. H.264/AVC has become a widely accepted coding standard and is expected that many of the content will be available in this format soon. This paper proposes a generic model of video summarization especially suitable for generating summaries of H.264/AVC bitstreams in a highly efficient manner, using the concept of temporal scalability via hierarchical prediction structures. Along with the model, specific examples of summarization techniques are given to prove the utility of the model.

Keywords: *video summarization, temporal scalability, H.264/AVC, content-based browsing*

1 Introduction

The need for effectively browsing the huge amount of visual content available in multimedia databases motivates the use of video summaries, in order to be able to preview the content that will be further used (accessed, deleted, archived, etc.). Video summarization includes a number of techniques exploiting the temporal redundancy of video frames of the video based on content understanding: these techniques try to provide the user with a compact representation of the content, but preserving those parts more “semantically” relevant[1-3]. Thus, the user can get an idea of what happens in the video without having to visualize the whole sequence. A compact and fast browsable representation is key in video search and retrieval applications, such as the access to large video databases or video-on-demand applications. In general, a summarized sequence is built from the source sequence selecting frames according to some kind of semantic analysis of the content.

A widely used representation for video summarization is the image storyboard, which abstracts the sequence in a set of keyframes. Many algorithms have been proposed for keyframe selection, using different criteria and abstraction levels[4, 5]. At a low level, keyframe selection can be formulated as an optimization problem where the distance between the summary set and the whole set of frames in the sequence is minimized[6]. At a higher semantic level, the sequence can be structured as a collection of shots, grouped into more abstract units (scenes), obtaining a hierarchical representation of the content[3]. With video content, sometimes it is more useful and meaningful for the user to present the content as segments of frames, instead of single frames. Segments provide information about the temporal evolution of the sequence. This representation is usually known as video skim, where significant segments are extracted from the sequence. A number of approaches have been used in video skimming, including visual attention[7], rate-distortion[8], graph modelling[9], image and audio analysis[10, 11] and high level semantics[4]. Between selecting single frames and selecting whole segments, there is still the possibility of selecting a variable amount of frames per segment, ranging from single frames to whole segments. Fast forwarding the sequence at a constant rate can provide the user with a preview of the content useful to browse it in a shorter time[12]. However, there are often less important parts that can be sped up, while more significant parts can be slowed down, having the same effect as using a variable fast forward guided by a semantic clue[13]. A similar technique using the same idea of variable sampling frames at a variable rate is semantic frame dropping, where some frames are discarded preserving the duration and timing of the source sequence. It is useful for video adaptation, as it discards frames from less important parts, being useful for adaptation in constrained environments[14] and in video transmission[15].

H.264/AVC is currently the video coding standard with best performance in terms of coding efficiency[16], which makes it very suitable for applications requiring the storage of high resolution or a large amount of content. Coding efficiency is also very desirable when the content is retrieved through a bandwidth constrained network. As a generic video coding standard with defined specific profiles and levels, H.264/AVC is flexible enough to be used in a broad spectrum of applications using digital video, ranging from video conferencing to high definition TV. In contrast with previous standards since the successful MPEG-2, H.264/AVC is receiving an increasing attention from industry and many video applications and systems are being developed using H.264/AVC as coding format[17]. Moreover, the standardization of new extensions for Scalable Video Coding[18] and Multiview Video Coding[19] include new functionalities into H.264/AVC that enable the development of enhanced applications. The hierarchical prediction structures are one of such tools, enabling not only improved coding efficiency but also temporal scalability.

Both video summarization and video coding are key aspects for a video retrieval system. Usually video summarization works on uncompressed domain, decoupled of the coding layer, but it leads sometimes to inefficient applications. There are some works that tackle this adaptation in the bitstream domain using the Bitstream Syntax Description (BSD) tools[20] of MPEG-21 Digital

Item Adaptation (DIA), aimed to generic adaptation of coded sequences directly operating with the bitstream. Particularly, the adaptation of H.264/AVC along the temporal axis using MPEG-21 BSDL is detailed in [21]. Gang *et al.*[14] describe a system using frame dropping based on the perceived motion energy for video adaptation. However, most of these works are used from the adaptation point of view, where the bitstream extraction is guided by constraints in the usage environment. If we consider video summaries as *semantic* constraints to be applied to the source sequence, we can use the same framework for summarization. This is the idea that motivates the work in this paper.

In this paper we propose to exploit coding structures and bitstream extraction not just for media-level (that is, content agnostic) adaptation, but for efficient summary generation. Taking advantage of the concepts of hierarchical coding structures and temporal scalability, the paper also proposes a specific organization of the bitstream suitable for summarization and a generic and flexible model for the representation of the results of summarization algorithms. With this approach, the generation of video summaries is efficiently solved using a bitstream extractor. The paper also describes how some common summarization techniques can take advantage of the proposed model, and presents experimental results which are compared to a non-hierarchical approach using MPEG-2.

The rest of the paper is organized as follows. Section 2 briefly overviews H.264/AVC and hierarchical prediction structures. Then, Section 3 describes the proposed summarization scheme and the differences with the conventional approach. In Section 4, some formal definitions and concepts are introduced to describe the structure of the bitstream in the model, and Section 5 describes the generation of the summary from this structured bitstream. Section 6 shows experimental results from several examples of summarization techniques, and finally Section 7 concludes the paper.

2 H.264/AVC and hierarchical prediction structures

As most of the preceding video coding standards such as H.262/MPEG-2[22] and H.263[23], H.264/AVC is based on the widely adopted model of hybrid block-based temporal prediction with block-based transform coding. It uses the traditional types of slices, I, P and B, with intra prediction and inter frame prediction. However, it includes new sets of coding tools which help to exploit better the redundancies in the sequence and to significantly improve the coding efficiency.

The recommendation specifies two different layers: a Video Coding Layer (VCL) which deals with the efficient representation of the samples and video content, and a Network Abstraction Layer (NAL) which deals with the format and header information in a suitable manner to be used by a variety of network environments and storage media.

A H.264/AVC bitstream is composed by a succession of NAL Units (NALUs), each of them containing a syntax structure. Some of such structures are the parameter sets. In particular, the Sequence Parameter Set (SPS and the Picture Parameter Set (PPS) are essential for the decoder in order to be able to reconstruct the sequence. They convey important header information such as frame resolution, profile, frame rate, etc. The information present in these parameter sets is used by the decoder to decode all the pictures following them, although it is not fixed and can be modified during coding if necessary, sending new parameter sets.

In H.264/AVC, a picture (frame or field) is divided into slices and represented by a set of them. Each slice contains a number of macroblocks and it is usually packed in a NALU. For the sake of simplicity, we will assume frame coding and one slice per frame, using interchangeably frame, picture, and slice along the paper. The pictures are also structured in Groups of Pictures (GOPs) related by temporal prediction, though pictures from different GOPs can be also related by prediction.

One of the key features of H.264/AVC to increase the compression performance is the possibility of specifying much more flexible prediction structures. In prior standards, there is a strict dependency between the ordering of pictures for motion compensation prediction (coding order) and the ordering for presentation. For instance, in MPEG-1/2/4, P pictures are predicted only from the preceding I or P picture, and B pictures are not used as references and are predicted only from the preceding and succeeding I or P pictures (see Fig. 1a). In contrast, in H.264/AVC any picture can be marked as reference and used for prediction of subsequent pictures. One of such family of prediction structures are the hierarchical prediction structures, where a set of pictures are coded at a base level (level 0) and at each step a new set of pictures is coded using previously coded pictures. The process is repeated with additional sets of pictures adding new levels in the hierarchy. In [24], the impact of hierarchical prediction in the coding efficiency is studied and experimental results showed that these structures have a good coding efficiency, which usually improves when the length of the structure is increased. Hierarchical prediction implicitly generates temporal scalable bitstreams. Each set of pictures from each temporal level form a new enhancement layer. Thus, selecting only the part of the bitstream corresponding to the pictures of the base layer, a coarse temporal resolution version of the bitstream can be decoded, that can be refined adding enhancement layers. Some typical hierarchical structures are those using dyadic decompositions, where the number of pictures is doubled with each enhancement layer. Fig. 1b and Fig. 1c show two typical hierarchical structures with 4 dyadic stages (3 enhancement layers):

- $I^0P^3P^2P^3P^1P^3P^2P^3$ (Fig. 1b). This structure does not use backward prediction, being compliant with the H.264 baseline profile, as only I and P pictures are necessary. All the predictions are from past pictures in display order, and thus the structural delay is 0 pictures.
- $I^0B^3B^2B^3B^1B^3B^2B^3(I^0)$ (Fig. 1c). This structure uses also backward prediction and B pictures will be necessary. In this case, the structural delay is 8 pictures but the coding efficiency is higher than in the previous structure.

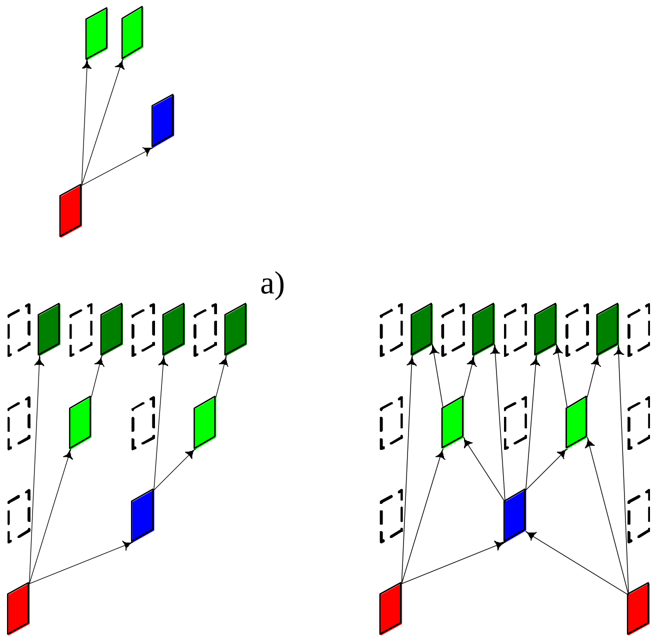


Fig. 1. Prediction structures: a) MPEG-2 structure, b) hierarchical structure in H.264/AVC using P pictures (structural delay 0), c) hierarchical structure in H.264/AVC using B pictures (structural delay 8)

3 Proposed model for summarization

In a wide sense, video summarization usually refers to a number of techniques which analyze at some level the semantics of the original sequence and generate a summary trying to cover most of these semantics, but that can be browsed much faster. In this paper we separate the whole summarization step into two stages: analysis of the input bitstream and generation of the summarized bitstream. A summary usually has the form of a set of images or a shorter video sequence. If the content of the frames (the pixels) is not changed by the summarization process, the summary generation consist of a simple selection of the frames and the subsequent reencoding into a suitable format (see Fig. 2a). The output of the analysis stage can be seen as a set of numbers with the indices of the keyframes to be included in the summary.

a)

b)

c)

Fig. 2. Methods for the generation of summaries: a) conventional approach, b) proposed approach for H.264/AVC bitstreams with online summarization analysis, c) proposed approach for H.264/AVC driven by metadata

The generation of the output summary requires the decoding of the frames to be included and the subsequent encoding stage. Usually, a high proportion of the frames are encoded predictively from previous frames, so all these frames must be also decoded previously to be able to decode a given frame. The whole transcoding process may have an important computational cost, especially when the summary is in the form of a video sequence (e.g. video skim).
However, if the bitstream is encoded in such a way that the transcoding process can be replaced by a simple selection of parts of the input bitstream, the generation of the summarized bitstream will be much more efficient. Scalable video codecs are very suitable for this approach as they use

a coding model that enables efficient adaptation by bitstream extraction. Particularly H.264/AVC can generate temporal scalable bitstreams using hierarchical prediction structures.

In order to be able to efficiently generate video summaries, we use H.264/AVC and hierarchical prediction structures with a different model to signal the output of summarization algorithms. This model relies on adapting the summarization algorithms to select groups of frames, related by the coding structure, and temporal levels, rather than single frames. Thus, once the analysis has been performed and the relevant groups of frames have been determined, the generation of the summary is much simpler and it consists of the selection of these groups of frames directly from the coded bitstream.

In the conventional summarization model of Fig. 2a, the output of the algorithm is the set of frame indices corresponding to the frames that should be included in the adapted sequence. Different video summarization algorithms will differ on the way the frames are selected, but the final result for each frame is that it is either included or not.

However, in the bitstream, the frames are coded in groups rather than individually. So it is more convenient to refer the output of the summarization algorithm to these groups rather than to single frames. We will call skimming curve the result of expressing the set of keyframes in terms of number of frames selected per each of these groups. Each of these groups is the main summarization unit in the proposed model and we will call them Skimming Units (SUs; a more formal definition will be given further on), and may be composed by frames from one or more GOPs. The hierarchical prediction structures enable the selection of different number of frames from each skimming unit. The model relies on the assumption that the only selection of frames allowed in each SU is the selection of those frames belonging to the same temporal level, and the selection of all the frames in that level. With this assumption, it is possible to translate the selection of single frames to a more suitable model that is the selection of SUs and a temporal level within each unit. We call this information Skimming Parameters, and they are the only information that the bitstream extractor needs to generate the summary. The architecture with the main modules is shown in Fig. 2b. The analysis for summarization is completely detached from the generation, and it would even be possible an architecture where analysis is performed previously (e.g. at encoding time) and the results are stored as metadata (see Fig. 2c). It is useful for storing several summaries as skimming parameters (possibly each one coming from different

analysis algorithms or modalities), while keeping only the input bitstream. The different available summarized bitstreams are then generated when each of them is required.
Selecting subsets of frames from the SUs rather than individual frames has the drawback of losing the exact position of each frame in time, and thus it is not possible to select a given frame of the bitstream, but it is possible to select a neighbour frame within the SU. However, as the frames are very similar to their neighbours (except when a shot change occurs), if the length of the skimming unit is small enough and the analysis stage is designed to prevent from including problematic units, for most applications there is almost no noticeable difference.

4 Skimming units

4.1 Basic definitions

If we consider a source temporal scalable sequence S_{source} with N frames and T temporal enhancement layers (that is, T+1 temporal levels), then the frame index can be notated as $n \in \Gamma = \{0,1,\ldots,N-1\}$ and the temporal level as $t \in \{0,1,\ldots,T\}$. For each temporal level *t*, a subset of frame indexes $\Gamma^t \subseteq \Gamma$ is defined, which depends on the coding structure (see Fig. 3), and the subsampled sequence S^t_{source} at that level. At the highest temporal level *T*, it satisfies $\Gamma^T \equiv \Gamma$ and $S^T_{source} = S_{source}$.

Definition 1. (Frame) Each frame in the sequence S_{source} can be notated as F^t_n where $n \in \Gamma^t$ is the index of the frame (in display order) and *t* is the number of the temporal layer at which it was encoded, and the minimum temporal level which it belongs to. Note that each temporal level inherits all the frames from previous levels, so F^t_n satisfies

$$F^t_n = F^{t+1}_n = \ldots = F^T_n, n \in \Gamma^t \qquad (1)$$

The type of slices is not considered in this definition, so, depending on the coding structure, F^t_n can refer to a frame coded with any I, P or B slices.

Definition 2. (Summary) A summary *S* with arbitrary frame indices $\Gamma_{SUMMARY}$ can be constructed as

$$S(\Gamma_{SUMMARY}) = \left(F_{k_0}^T, F_{k_1}^T, \ldots, F_{k_i}^T, \ldots, F_{k_{W-1}}^T\right), k_0 < k_1 < \ldots < k_i < \ldots < k_{W-1}, k_i \in \Gamma_{SUMMARY} \quad (2)$$

where W is the length of the summary in frames. Considering the source sequence as a special case of summary $S = S_{source}$, it will satisfy $\Gamma_{SUMMARY} = \Gamma$. Similarly, the source sequence downsampled at temporal level t can be obtained with $\Gamma_{SUMMARY} = \Gamma^t$

Definition 3. (Skimming unit) A skimming unit G^t is a set of consecutive frames (in display order) at level t related by the prediction structure and that can be decoded independently from the other frames in the sequence

$$G^t = \left(F_{k_0}^t, F_{k_1}^t, \ldots, F_{k_i}^t, \ldots, F_{k_{L^t-1}}^t\right) \quad (3)$$

where $F_{k_i}^t \in S, k_0 < k_1 < \ldots < k_i < \ldots < k_{L^t-1}, k_i \in \Gamma^t$, and L^t is the length of the SU. The basic idea behind the concept of SU is depicted in Fig. 3.

The aim of this model is to get a partition of the sequence into a set of SUs in such a way that the sequence can be obtained as:

$$S_{source} = \left(G_0^T \vdots G_1^T \vdots \cdots \vdots G_m^T \vdots \cdots \vdots G_{M-1}^T\right) \quad (4)$$

where M is the number of SUs in the partition. Due to the hierarchical structure, at each temporal level t, it is possible to construct another partition that satisfies

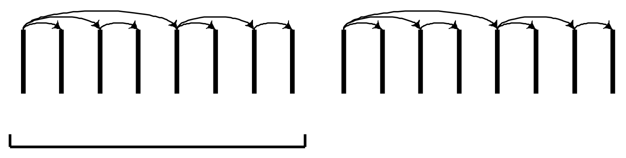
$$S_{source}^t = \left(G_0^t \vdots G_1^t \vdots \cdots \vdots G_m^t \vdots \cdots \vdots G_{M-1}^t\right) \quad (5)$$

with $G_m^t \subset G_m^T, m = 0,1,\ldots,m,\ldots,M-1$. In general, the SUs in the partitions satisfy

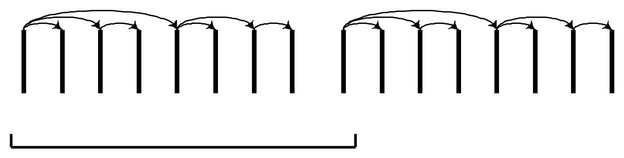
$$G_m^0 \subset G_m^1 \subset \ldots \subset G_m^t \subset \ldots \subset G_m^T, m = 0,1,\ldots,m,\ldots,M-1 \quad (6)$$

Definition 4. (Overlapped skimming units) We say that two skimming units G_i^t and G_j^t are overlapped if they share one or more frames, that is,

$$G_i^t \cap G_j^t \neq \varnothing, i \neq j \quad (7)$$



a)



b)

Fig. 3. Examples of SUs: a) low delay structure with P frames, b) overlapped SUs using hierarchical B pictures (display order).

An example of overlapped SUs is shown in Fig. 3b. Note that, using (1) in a set of partitions of the type described in (6), if G_i^t and G_j^t are overlapped, $G_i^{t'}$ and $G_j^{t'}$ are also overlapped with $t \le t' \le T$. However it does not imply the opposite: in Fig. 3b, all the SUs of levels from 1 to 3 are overlapped, but the SUs at level 0 are not.

Formally, expression (4) is not valid with overlapped SUs, since if there are some frames belonging to different SUs, there will appear replicated in the reconstructed sequence. In some cases, as in the structure of Fig. 3b, it is still possible to reconstruct the source sequence from a set of overlapped SUs. Assuming overlapping only with the previous SU, the source sequence at level *t* can be obtained as

$$S_{source}^{t} = \left(G_0^t \vdots G_1^t \setminus G_0^t \vdots \cdots \vdots G_m^t \setminus G_{m-1}^t \vdots \cdots \vdots G_{M-1}^t \setminus G_{M-2}^t \right) \tag{8}$$

where \ is the set difference operation.

4.2 Skimming units and coding structures in H.264/AVC

As previously discussed, in this analysis we will focus on hierarchical structures and particularly on some common structures, discussing the effects of the coding structure on delay, compression and the type of SUs in the proposed summary generation model. More complex coding structures can be used to improve the coding efficiency and to restrict the structural decoding delay.
The most simple of the considered structures is the low delay structure using only P pictures with forward prediction. An example of length 8 of this structure is the dyadic decomposition $I^0P^3P^2P^3P^1P^3P^2P^3$, shown in Fig. 3a. The most important characteristic of this structure is that it does not require the use of B pictures. As a first advantage, it can be decoded by a baseline profile decoder. A second advantage of this structure is its suitability for low delay applications, having a structural delay of 0 pictures. And regarding to the proposed model of SUs, it enables the structuring of the bitstream in non-overlapped SUs, which is very desirable. However, the main drawback of this structure is the lower compression efficiency comparing to structures using B pictures. Another drawback is the appearing of temporal artefacts due to the different prediction paths followed to code each picture of the structure[18], accumulating prediction errors in a non-equal manner.
On the other hand, adding backward and bidirectional prediction to the coding structure increases notably the compression efficiency, and reduces significantly temporal artefacts. It is achieved using B pictures in the prediction structure. A typical example of this structure providing the same functionality as Fig. 3a in terms of temporal scalability is $I^0B^3B^2B^3B^1B^3B^2B^3(I^0)$ (see Fig. 3b), where the length of the GOP is also of 8 pictures, but also needs an additional I picture from an adjacent GOP. Adding bidirectional prediction has effects on the coding order increasing the

structural delay: the pictures used as references for backward prediction must be coded before the pictures referencing them. Fig. 4 shows the coding order of the pictures from previous example. The two I pictures must be coded before all the pictures between them, leading to a structural delay of 8 pictures. The SUs of structures using B pictures can be overlapped when they share references from different GOPs. Most of the resulting SUs from the previous example are always overlapped, except for temporal level 0, where no prediction is used.

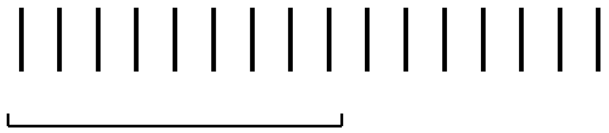


Fig. 4. Skimming units using hierarchical B-pictures (coding order).

Compression efficiency can be further improved using prediction in the base layer instead of coding the pictures only as I slices. The structure of Fig. 5 is the result of replacing a number of I pictures in the base layer of Fig. 3b by P pictures, which are predicted from the previous I or P picture in the base layer. Note that it will increase the coding efficiency, but it will be also increase the length of the SU. A larger SU brings a decreasing of the precision in selecting single pictures at a given instant. There is always a trade-off between coding efficiency with long SUs and precision in selecting frames using this model.

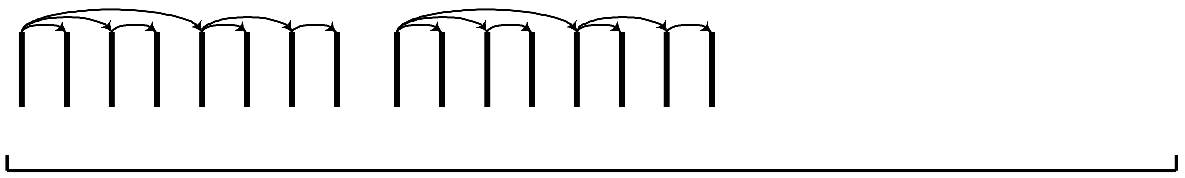


Fig. 5. Skimming unit using long term prediction for improved compression efficiency.

4.3 Other coding structures

In this paper we have focused on an important family of coding structures (hierarchical structures) providing good performance in terms of compression and that can be used to generate summaries within the proposed model. For simplicity, we describe the case of sequences with constant SU length, although it can be extended to variable SU length. In this last case, as analysis results must be aligned with the partition in non constant length SUs, it usually makes more complex both analysis and generation.

Note that typical non hierarchical structures could be used, providing a two layers approach (i.e. only intra frames for lower frame rate and the whole GoP for full frame rate) and even three layers if B frames are used, although hierarchical structures are much more flexible.

H.264/AVC standard is even more flexible, providing new tools such as using predictions from multiple references in a same frame, or using arbitrary frames as references, and thus much more complex coding structures can be used. Note that not all the possible coding structures in the standard are suitable for the proposed summarization model, as it is not always possible to define proper SUs, decodable independently from the others.

5 Generation of the summary

5.1 Bitstream extraction

In the proposed approach, the generation of the output summary consists of the composition of SUs guided by the results of the summarization analysis. Fig. 6 shows the process of generation of the output summary. Given a SU index $m = 0,1,\ldots,M-1$ and the corresponding set of available SUs in the bitstream $\Omega_m = \{\varnothing, G_m^0, G_m^1, \ldots, G_m^t, \ldots, G_m^T\}$, only one of the SUs in Ω_m is included in the output summary. Note the presence of the empty set $\varnothing$ as a possible choice, meaning that no SU is included at index *m*.

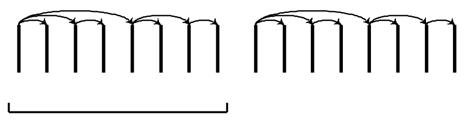


Fig. 6. Bitstream adaptation guided by skimming parameters.

The parameters that guide the inclusion or not of a SU and its temporal level are the Skimming Parameters:

Definition 5. (Skimming parameters) Let be the pair $p_m = (b_m, l_m)$, then the skimming parameters are defined as a sequence of pairs

$$P = \{p_m : p_m = (b_m, l_m), m = 0,1,\ldots,M-1\} \tag{9}$$

where $b_m \in \{0,1\}$ is a boolean variable meaning the no inclusion of any SU at index *m* when b_m=0, and the inclusion of the skimming unit $G_m^{l_m}$ at temporal level $l_m \in \{0,1,\ldots,T\}$ when b_m=1.

Given the skimming parameters $P_{SUMMARY}$, obtained from the analysis, the output summary is built as

$$\tilde{S}(P_{SUMMARY}) = \left(\tilde{G}_0 \vdots \tilde{G}_1 \setminus \tilde{G}_0 \vdots \cdots \vdots \tilde{G}_m \setminus \tilde{G}_{m-1} \vdots \cdots \vdots \tilde{G}_{M-1} \setminus \tilde{G}_{M-2}\right) \tag{10}$$

where $\tilde{G}_m$ is obtained as

$$\tilde{G}_m = \tilde{G}_m(p_m) = \begin{cases} G_m^{l_m} & b_m = 1 \\ \varnothing & b_m = 0 \end{cases}, p_m \in P_{SUMMARY} \tag{11}$$

Note that (11) provides an alternative to (2) for summary generation, taking advantage of the hierarchical structures. The possible summaries generated using SUs are a subset of the possible summaries generated using arbitrary frames, as the former are constrained by the coding structure. Nevertheless, this constraint is not very important in practical applications using common hierarchical coding structures with reasonable SU length and the analysis designed conveniently with the model of generation. Designing the analysis algorithm properly and adapted to this model helps to avoid possible artefacts. For instance, the analysis algorithm can detect shot boundaries in order to avoid the inclusion of those SUs having frames from different shots, which can produce artefacts in some cases.

Although bitstream extraction consists basically of the copy of chunks of the input bitstream to the output bitstream, sometimes some header modifications may be necessary in order to conform a valid bitstream. In H.264/AVC, the Decoded Picture Buffer (DPB) stores decoded pictures that may be used as references by following slices. The indices of the references signalled in the header are referred to the current status of the DPB, according to the decoding process, that includes and discards pictures from the buffer dynamically. The proposed method introduces discontinuities in the status of the DPB when some SUs are discarded. If the first slice (in coding order) of each SU is an IDR (Instant Decoding Refresh) slice, the DPB is flushed. Thus, the DPB at the beginning of each SU is always empty. However, if I slices are used instead, the DPB is not flushed, so its header must be modified and converted to an IDR slice to reset the status of the DPB. The next slices may also need to be updated conveniently according to the new status.

5.2 Presentation of the summary

The pictures obtained with (11) form a sequence of pictures, but without timing information in terms of presentation delays between successive frames. Often, the presentation of the frames in the terminal varies depending on the summarization technique. The way they are presented in the terminal can be modified in order to obtain a different effect. For instance, it can be used to control the delay between slides if the set of frames in the summary is presented as a slideshow. Thus, the possibility of adjusting the presentation of the frames is also a desirable feature and it should be also considered in the generation of the summary.

We consider two presentation schemes: constant frame rate and variable frame rate. In a constant frame rate presentation, each picture is presented after the previous one with the same constant delay. On the other hand, if the frame rate varies throughout the sequence, the delay between pictures also varies, and must be signalled in the bitstream. Constant frame rate does not require any special processing, unless specifying a different frame rate, if necessary, at the beginning of the summarized bitstream.

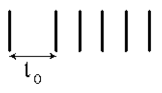


Fig. 7. Constant and variable frame rate presentation schemes.

Although variable frame rate can be managed at the system layer, we propose two methods to achieve such variable frame rate behaviour at the coding layer (see Fig. 7):

1. Signalling the changes of frame rate. A SPS is required to be sent before each picture with a frame rate different from the previous picture. The value of `num_units_in_tick` and `time_scale` is set according with the frame rate. Note that only a single SPS modified dynamically is necessary and the identifier `seq_parameter_set_id` in the slice header is not changed. This method can fit to a larger variation of frame rates.
2. Using a predefined set of SPSs. If the possible frame rates are restricted to a fixed number of possibilities *K*, it would be more useful to define *K* SPSs with different values of `num_units_in_tick` and `time_scale` and include them at the init of the bitstream, with `seq_parameter_set_id` varying from 0 to *K*-1. It will be necessary to send *K* PPSs, each of them referring to one of the SPSs using `pic_parameter_set_id=seq_parameter_set_id` from 0 to *K*-1. In each header slice, the value of `pic_parameter_set_id` needs to be modified to use the corresponding SPS via the PPS.

6 Experiments

In order to demonstrate the utility of this model to generate video summaries, this section describes the application of the proposed framework in examples of different summarization techniques, and how they can be adapted to the proposed representation model. We describe some simple and widely used summarization techniques, which have been adapted to fit into the proposed framework and representation model. It must be noted that the analysis itself is outside of the scope of this paper, which is focused on the representation and generation of the summaries. More complex and specific content-based analysis algorithms could be used to obtain better summaries in terms of semantic coverage. Along with the study of these different modalities, some subjective and performance evaluations are also included.

The algorithms were tested with the sequence *Sun-Earth Connection*, a video downloaded from the Open Video Project's repository[25] in MPEG-2 format, and transcoded to H.264/AVC using the JM 12.4 reference implementation. The sequence has 11593 frames of 720x480 pixels at 30 frames per second. In order to study the influence of the GoP length and frame type, the sequence was encoded with different hierarchical structures, which differ in the GoP length (from 1 to 32 frames) and in the use of either P or B slices. In the experiments we assume a base layer with only I slices, so each GoP corresponds to a SU. The use of B slices leads to overlapped SUs.

In order to have a comparison with a coding scheme not using the flexible hierarchical prediction structures of H.264/AVC, we have also implemented the generation of video skims and storyboards for MPEG-2 coded sequences, using bitstream extraction. For storyboards only the selected I frames are preserved in the output bitstream, and for video skims complete GoPs are preserved. In order to compare the systems under the same conditions, the test sequence was reencoded in MPEG-2 with the same GoP lengths as for H.264/AVC and for two GoP structures: one with only P frames (IPPP…) and another with P and B frames (IBPB…). The summaries are generated using the same skimming parameters as those used for H.264/AVC.

6.1 Summarization modalities

Once the proposed model has been presented, we can introduce and characterize several types of summarization modalities. Although more complex combinations could be possible, depending on the way the selection (from single frames to complete skimming units) is done and the summary is presented (e.g., collection of images, video fast preview) we will distinguish four types of summarization modalities and their corresponding skimming curves (see Fig. 8). These modalities can also be described by the values that the skimming parameters $p_m = (b_m, l_m)$ take:

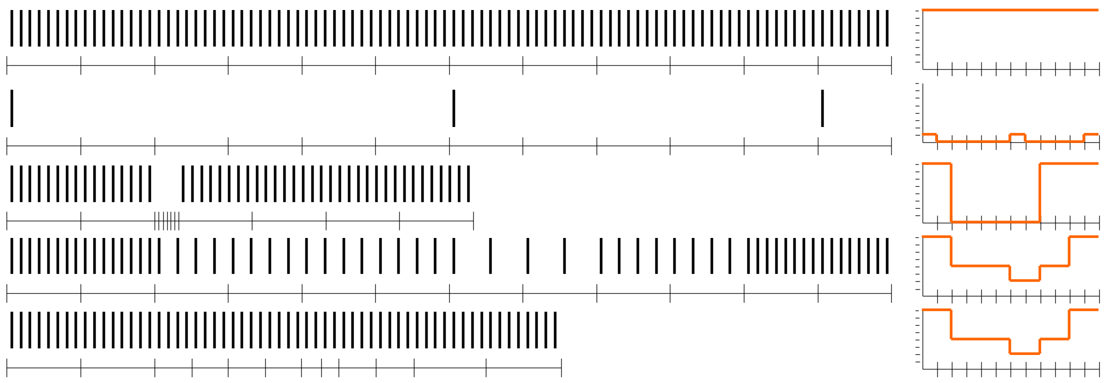


Fig. 8. Frame selection (left) and skimming curve (right) for different types of video summarization.

- Image storyboard: the skimming curve is zero (no frames) for each SU, except in those where there is a keyframe. In practical applications, no more than one frame per SU would be necessary, for SUs with reasonable length. Then the value of b_m is either 0 or 1, meaning that there is a keyframe in the *m*-th SU, the l_m=0, selecting a single picture. Thus, the SU included in a storyboard is of the form G_m^0 .
- Video skim: the adapted sequence is shorter than the input sequence, obtained by selecting some segments of the input sequence (e.g. a trailer of a movie). In this case, a typical skimming curve can be easily obtained having either all the frames in the skimming unit or either none. In the model it means that full SUs are either included or excluded (b_m either 0 or 1), and l_m=T always. Thus, the SU included in a storyboard is of the form G_m^T . The presentation scheme is at the source frame rate ($\tau_m = \tau_{SOURCE}$).
- Frame dropping: the number of frames of each SU varies according to some analysis of the semantic or perceptual relevance of the frames, dropping those considered less relevant than the others. There are no constraints on the values of b_m or l_m, but the duration of the sequence is preserved, following the timing of the source sequence. The interval between pictures varies with the temporal level l_m selected, trying to keep the duration of each SU constant. In the common case of dyadic structures the interval between frames can be computed as

$$\tau_m = \tau_{SOURCE} 2^{T-t} \qquad (12)$$

where t is the scale of the SU at index m.

- Fast preview: similarly to the previous approach, the frames are selected according to some perceptual or semantic criteria. However, the sequence is presented at a constant frame rate in a shorter duration. The number of pictures selected in each SU can take all the values, so there are no constraints on b_m or l_m. The pictures are presented with a constant frame rate, with a constant interval between pictures $\tau_m = \tau_{playback}$, usually the same as in the source sequence.

6.2 Image storyboard

Storyboards are represented by few independent frames, very dissimilar between them, trying to cover the semantics in the sequence. For this reason, a widely used criterion is the distance between frames in a feature space. Clustering algorithms have been successfully used to select few representative frames[27, 28] which build the storyboard summary. In order to prove the utility of the proposed framework in this case, a simple clustering is used based on K-means algorithm. Each frame is decoded into the YUV colour space and divided into 2x2 subimages in order to have some information (with an efficient feature) about the spatial distribution of colours what can not be done with only one global histogram. Each subimage is represented with a 32-bin histogram for the Y component and two 8-bin histograms for the U and V components. Each frame is then represented by a 192-D feature vector. The feature vectors from all the frames are clustered using the K-means algorithm with the Euclidean distance, resulting in K centroids. For each centroid, the closest frame from its cluster is selected as keyframe. Temporal subsampling is often used in summarization to reduce the computational burden without degrading the quality of the summaries[28], due to the redundancies between consecutive frames. This temporal subsampling can be achieved in the summarization model processing only the pictures at the lower temporal resolution of the bitstream (G_m^0). Besides, selecting the same temporal level in analysis and in adaptation prevents from problems derived from the lack of exact temporal localization in the model (e.g. when shot changes occur in a SU). The output of the algorithm for 10 clusters corresponds to the skimming parameters shown in Fig. 9.

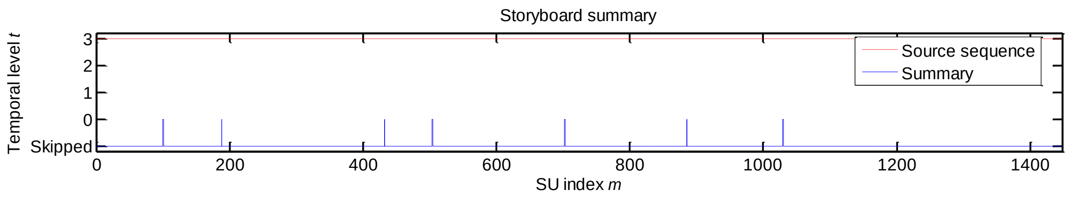


Fig. 9. Skimming parameters in the case of a 7 keyframes storyboard with a SU length of 8 frames.

The resulting summaries for different values of K are shown in Fig. 10.

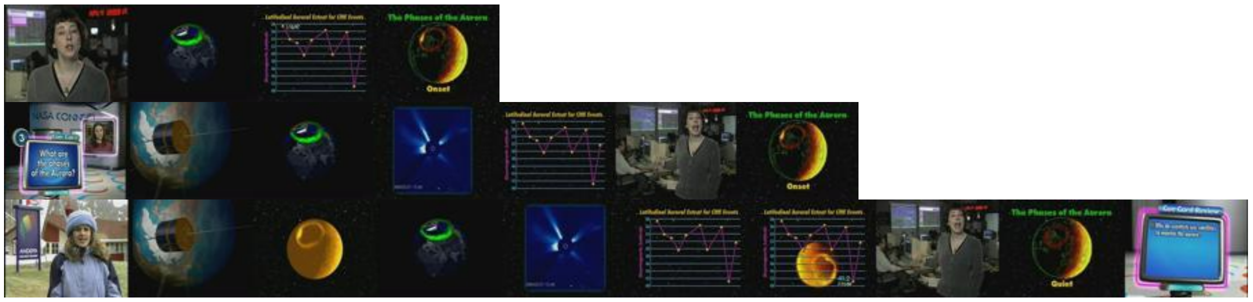
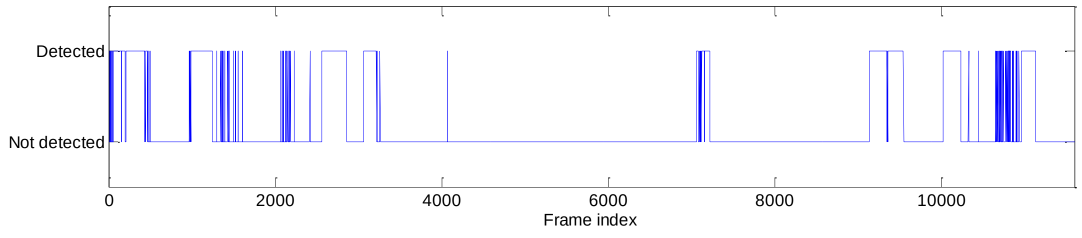


Fig. 10. Examples of storyboards with 4, 7 and 10 keyframes.

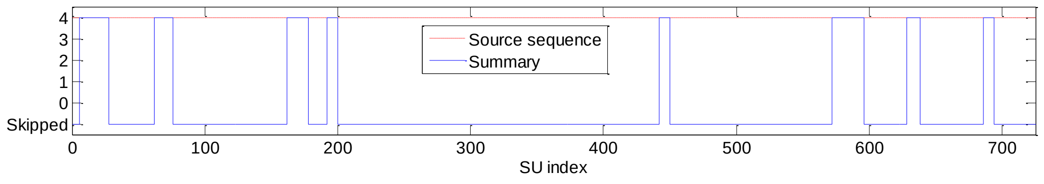
6.3 Video skimming

One of the most interesting summarization applications is the generation of video skims, simply selecting video segments according to some semantic criteria. Depending on the application, the selection technique can vary, for instance, from video trailers with the most active parts of the sequence to video skims with the parts with a given person. In this experiment, we assume that the segments with more semantic relevance are those with an anchorperson. This criterion may be useful in many domains, such as news programs.

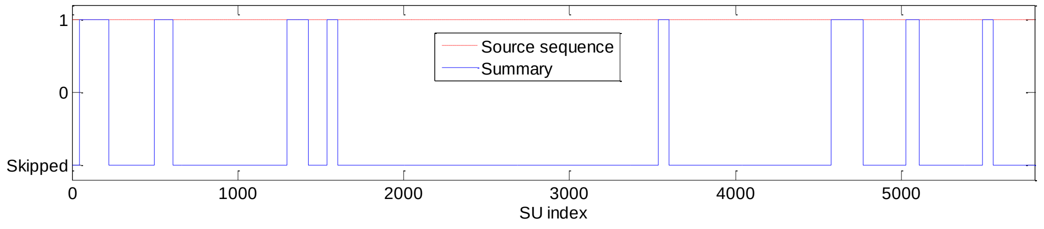
The face detection method of Viola and Jones[29] is used to mark the frames with at least one face detected (see Fig. 11a). The minimum size of the face is set to 88x72 pixels. If the number of frames with face detection in a SU is greater than the number of frames without any detection, the SU is marked to be included in the video skim. In order to cope with false detection and to reduce undesirable short segments or gaps, a moving median filter is used to obtain a smoother curve. In the experiments we used a window length of 11 for the median filter. Then, the higher level of each selected SU is included in the skim. Fig. 11b represents the skimming parameters obtained for 32 frames per GoP. As we use a dyadic decomposition, it is possible to build a summary with the same frames using a submultiple of 32 as GoP length, which is useful to compare the same summary generated with different GoP lengths. For example, Fig. 11c shows the equivalent skimming parameters of Fig. 11b for 2 frames per GoP.



a)



b)



c)

Fig. 11. Results for the video skim: a) frames with face detected, b) skimming parameters (32 frames GoP), c) skimming parameters (2 frames per GoP).

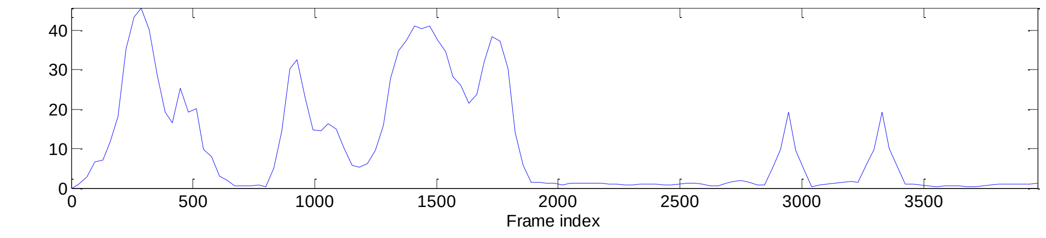
6.4 Fast preview

We demonstrate the usefulness of the model for generation of fast previews based on semantic clues using the method proposed in [30]. In this method, activity is used as semantic clue guiding the playback of the sequence. The temporal levels are selected according to this assumption in order to approach to the target frame rate. Skipping is also used to achieve a lower virtual frame rate along several SUs.

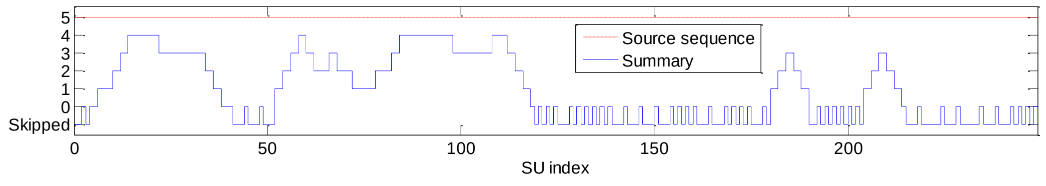
A widely used measure of activity is the MPEG-7 intensity of motion activity descriptor[31], which has been successfully used in indexing, fast browsing[13] and video rate control[15]. The basic assumption in [30] is that the instantaneous frame rate in the output sequence should be proportional to the measure of activity. We use a variant of the MPEG-7 intensity of motion activity without quantization (in order to have the flexibility provided by the use of a continuous range), and adapted to the variable block size motion vectors of H.264/AVC. An advantage is that it can be computed directly in the compressed domain with minimum cost, avoiding most of the decoding process. Besides, as the activity is computed in a SU basis, we use only the picture of the first enhancement level (P^1 or B^1, depending on the coding structure), without having to process the rest of the pictures of the SU. The activity I_m is obtained as

$$I_m = \sqrt{\frac{1}{A_{MB}} \sum_{i=1}^{N_{MV}} a_i \left\| \vec{m}_i - \overline{\vec{m}} \right\|^2} \qquad (13)$$

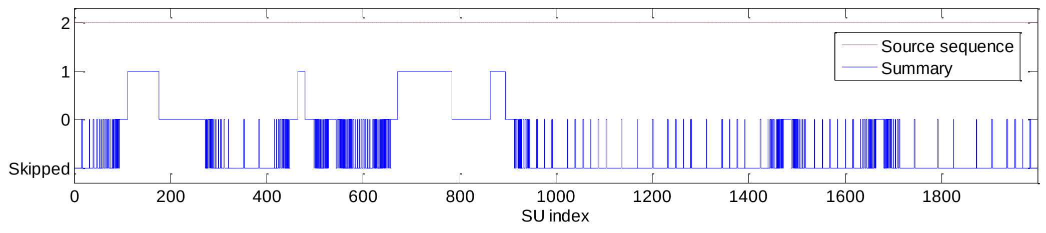
where N_{MV} is the number of motion vectors in the picture (including all the partitions of macroblocks), and $\vec{m}_i$ is each of the motion vectors. The parameters a_i, A_{MB} and $\overline{\vec{m}}$ are, respectively, the equivalent area of the partition of i, the total area (both measured in pixels) and the mean motion vector.



a)



b)



c)

Fig. 12. Details of the results for fast preview of the test sequence: a) filtered activity, b) skimming parameters (32 frames per GoP), c) skimming parameters (4 frames per GoP).

The activity curve obtained from (13) is very noisy and it can lead to temporal artefacts in the form of undesired sudden changes in the frame rate. Besides, there are a lot of isolated impulses, most of them due to shot changes. In order to reduce these effects, the curve was smoothed using a median filter (see Fig. 12a). The skimming parameters are then computed quantizing the activity curve in a logarithmic scale (note that the number of frames for each layer is proportional to 2^t) and using the skipping procedure to achieve less frames per SU when required. Fig. 12b and c represent the skimming parameters obtained for 32 and 4 frames per GoP, and they will lead to summaries with exactly the same frames. The output summary is then generated using constant frame rate for playback.

6.5 Semantic frame dropping

The previously described fast preview technique can be easily converted to a content-based frame dropping scheme. Actually, activity based frame dropping has been used previously in video transmission[15]. In contrast, a frame dropping scheme has the same timing as the original sequence. In the skimming model, it means that the pictures in each SU should be played at a variable frame rate, in such a way that it compensates the effect of the frame dropping.
In this experiment, in order to compare with the previous method, the curve of Fig. 12c is used to generate the output sequence. The variable frame rate is achieved using the approach of signalling each change in the frame rate with a new SPS. The value of `num_units_in_tick` is computed using (12) plus the total duration of the skipped SUs when one or more SUs are skipped. The bitstream is the same as in the case of fast preview, but with some additional bytes due to the parameter sets necessary to convey the information of variable frame rate.

6.6 Subjective evaluations

The first experiment studied the effect of the lack of accuracy in the selection of arbitrary frames when the length of the SU increases. The experiment tries to quantify the effect over the skims and storyboards obtained from the test sequence. Firstly, the sequence is analyzed with frame precision and a set of frames to be included is obtained at the finest scale (i.e. 1 frame per GoP). For each GoP length, a different set of skimming parameters is obtained by subsampling the previous set. For storyboards, the closest I frame of the corresponding SU at a given scale is selected. For skims, in a first approach (skim 50% in Fig. 13), a SU is included at a given scale if half the frames belonging to it are included in the summary at the finest scale.

Then, ten people were asked to evaluate the distortion that they perceived comparing to the summary obtained with frame precision. Two criteria are used: visual distortion, which measures if the subsampled summary is visually similar to the reference one and if annoying artefacts are included in it; and semantic distortion, which measures if the subsampled summary is equivalent to the reference one and if important information is lost or not useful information is included by the effect of the subsampling.

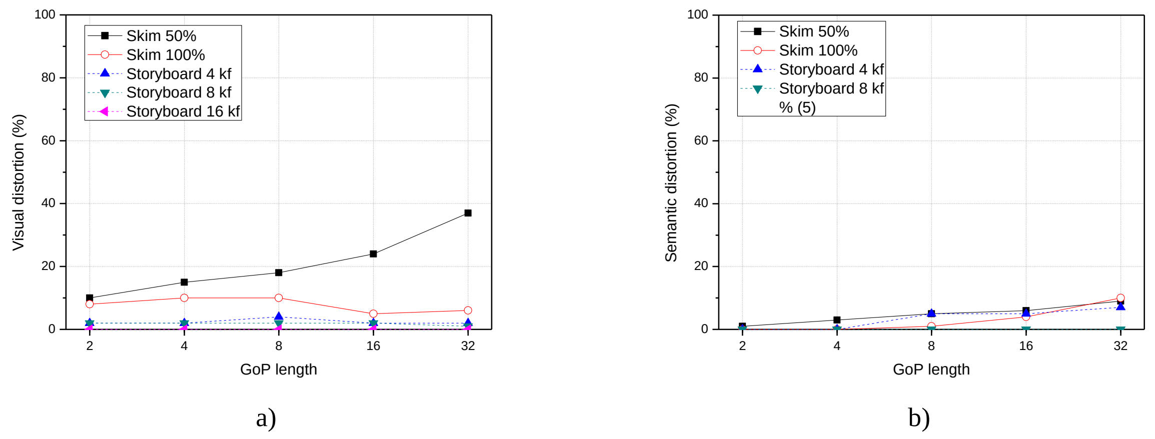


Fig. 13. Subjective evaluation results: a) visual distortion, b) semantic distortion.

In the case of storyboard, almost every subject agrees that no significant distortion is perceived for both visual and semantic points of view. In the case of skims, there is no semantic distortion perceived, as most of the information of the summary is still present in the summaries with coarser scales. However, visual distortion increases with GoP length, being important at 32 frames per GoP. The approach used for subsampling has the drawback of including new frames at the boundaries of previous segments, which leads to temporal artefacts, mainly when some frames from adjacent shots are included. However, this problem can be reduced if the approach used in analysis is designed carefully with the generation model in mind, in order to avoid in advance the inclusion of problematic SU (for example, SU including shot boundaries). In the experiment we also used a slightly different approach for subsampling, selecting a SU at a scale only if all the frames are included at the finest scale (skim 100% in Fig. 13). Now the problem is that some frames can be lost at the boundaries, but we avoid the problem of including new frames at the boundaries. Fig. 13 shows that this alternative subsampling approach can reduce significantly the visual distortion, and only a small semantic distortion is perceived (note that audio is not considered in the work reported in this paper, and a different approach would be probably necessary in that case).

6.7 Performance evaluation

Section 2 remarked the advantages and drawbacks of hierarchical structures using P or B pictures. GoP length and the use of B pictures can help to improve coding efficiency. However, they also can affect to the performance in the generation of the summary. This section provides some experimental measures in order to assess the performance of the system for both coding efficiency and processing time.

Firstly, the test sequence was encoded with the same configuration in both cases (quantization parameters, motion estimation parameters, etc.), except that one structure uses P pictures and the other uses B pictures. Fig. 14 shows the mean bitrate of the sequence coded using different GoP lengths and compared to the same sequences coded with MPEG-2. As expected, coding efficiency drops with GoP length increase, although, for this sequence, it does not improve for GoPs larger than 8 frames with H.264/AVC.

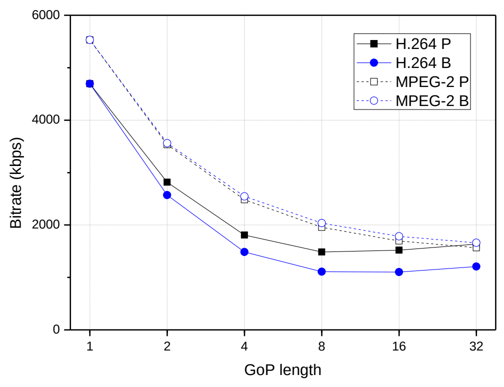


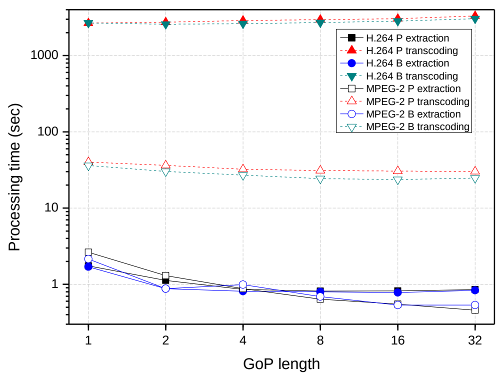
Fig. 14. Comparison of the coded sequences.

A major benefit of the use of bitstream extraction in the summarization process is the low computational cost and a faster generation of the bitstream. The time spent in the generation of the bitstream is compared to that spent using a conventional approach with a transcoder generating the same frames. The transcoders used in these experiments consist of simple cascade of decoder, extractor of frames (in uncompressed YUV) and encoder. The encoder uses the same configuration used for the encoding of the input bitstream. However, a comparison of different approaches in terms of processing time is highly dependent on the specific implementation of a codec and its degree of optimization. Table 1 provides complementary information to the experimental curves, pointing briefly some key aspects of the implementation of each module used in the experiments.

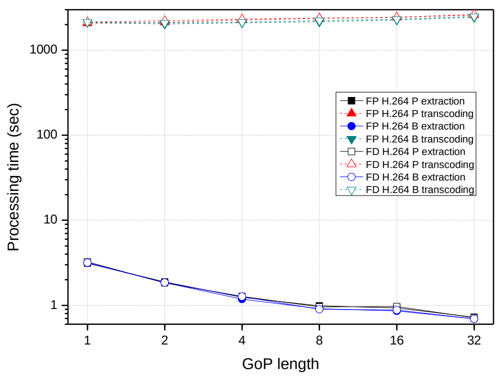
Module	Based on	Optimization	Uses bitstream description
Extractor H.264	Some parts of JM 12.4	Medium	Yes
Transcoder H.264	Decod+Encod JM 12.4	Very low	-
Extractor MPEG-2	-	Medium	No
Transcoder MPEG-2	FFmpeg	Medium-high	-

Table 1. Details of the software implementations used in the experiments.

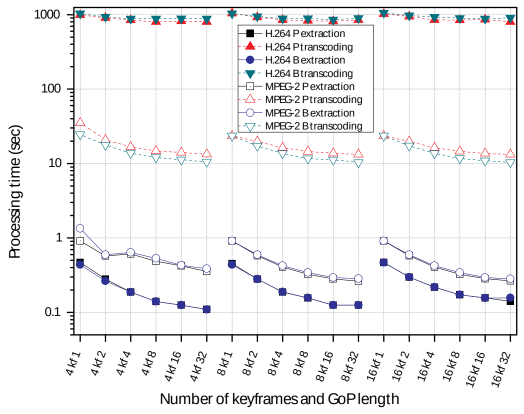
In contrast with the previously described subjective experiments, we must only compare the generation of the summarized bitstream, independently of the analysis and the GoP length. We perform the analysis at the coarsest scale (32 frames per GoP) and then reconstruct the equivalent skimming parameters to scales with higher accuracy (see Fig. 11b and c and Fig. 12b and c). Thus, the summary is fixed and it includes the same frames for all the variations.
Fig. 15 shows the results for video skim, storyboard, fast playback and frame dropping. In all of them, bitstream extraction performs significantly faster than transcoding for both H.264/AVC and MPEG-2, with a factor of about 50 and 1000. The use of P or B pictures does not affect significantly the performance in the tests.



a)



b)



b)

Fig. 15. Processing time: a) skim, b) storyboard c), fast playback and frame dropping.

As expected, summaries with more frames, such as skims, require more extraction time than storyboards, with fewer frames. In this last case, H.264/AVC extraction performs better than MPEG-2 extraction. This fact is due to the use of bitstream descriptions by the H.264/AVC extractor, which provides effective information about the localization and boundaries of the packets in the input bitstream. The bitstream description is generated by the encoder and stored with the bitstream. Thus, most of the header parsing can be avoided. In contrast, the MPEG-2 extractor does not use any extra information so it needs to parse each header in order to detect the boundaries of each packet.

Although the generation process is specified for each SU (or GoP), the basic unit of the bitstream is the NAL unit. In this sense, the generation time is approximately independent of the GoP length, as the number of NAL units does not vary significantly (although the relative amount of NAL units with I slices and NAL units with P/B slices does). However, the curves in 15 show that processing time decreases as GoP length increases. This fact is explained considering that the storage and parsing of skimming parameters by the extractors is not optimized for these experiments and consists of reading plain text files. Note that for shorter GoPs there are more skimming parameters than for longer GoPs, and thus with this implementation the parsing time of these files becomes more dominant as GoP length decreases.

In the case of fast playback and frame dropping (see 15b) there are no significant differences, as both lead to almost the same bitstream, differing only on the few additional packets of frame dropping.

The results show that extraction is considerably faster than transcoding. Although the performance of the implementations used in previous tests can be further improved for both transcoding and extraction (especially for H.264/AVC transcoding), and thus the processing time can be further reduced. However, bitstream extraction provides a simple way of adaptation which is intrinsically faster than transcoding, and in the same conditions, should outperform transcoding in terms of processing time.

7 Conclusions

A way to describe a bitstream specifically designed for summarization can be very useful to efficiently generate video summaries. In this paper we have proposed some concepts and a model for representing video summarization results taking advantage of coding structures and particularly of the hierarchical prediction structures of H.264/AVC. This representation enables a flexible and efficient generation of the bitstream using a bitstream extraction framework. The model has been tested with some application examples and used to simulate summaries with a real sequence. Even with simple cues such as motion activity or colour histograms, useful summaries can be obtained and presented in several modalities, such as fast previews, image storyboards or video skims. The proposed approach has been tested with different parameters and it has been proved to be significantly faster than an alternative approach based on transcoding. It was also tested with MPEG-2 and the results show that H.264/AVC with hierarchical prediction structures give better compression performance and can be used in a wider range of summarization modalities.

The proposed model is generic enough and independent of the analysis stage, so it can take advantage of more complex analysis and abstraction algorithms, using higher semantic levels to provide efficient access to meaningful summaries. Future work will focus on improving the analysis and summarization algorithms including additional visual cues.

8 References

[1] S.-F. Chang and A. Vetro, "Video adaptation: concepts, technologies, and open issues," *Proceedings of the IEEE,* vol. 93, pp. 148-158, Jan 2005.

[2] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video-content analysis and retrieval," *Multimedia, IEEE,* vol. 9, pp. 42-55, July-Sept. 2002.

[3] X. Zhu, A. K. Elmagarmid, X. Xue, L. Wu, and A. C. Catlin, "InsightVideo: toward hierarchical video content organization for efficient browsing, summarization and retrieval," *Multimedia, IEEE Transactions on*, vol. 7, pp. 648-666, Aug. 2005.

[4] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *Journal Of Visual Communication And Image Representation,* vol. 7, pp. 345-353, December 1996.

[5] M. M. Yeung and B.-L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 7, pp. 771-785, Oct. 1997.

[6] H. S. Chang, S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 9, pp. 1269-1279, Dec. 1999.

[7] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhang, "A generic framework of user attention model and its application in video summarization," *Multimedia, IEEE Transactions on,* vol. 7, pp. 907-919, Oct. 2005.

[8] Z. Li, G. M. Schuster, A. K. Katsaggelos, and B. Gandhi, "Rate-distortion optimal video summary generation," *IEEE Transactions on Image Processing,* vol. 14, pp. 1550-1560, Oct. 2005.

[9] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Automatic video summarization by graph modeling," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 2003, pp. 104-109.

[10] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding," in *Content-Based Access of Image and Video Database. Proceedings of IEEE International Workshop on*, 1998, pp. 61-70.

[11] Y. Li, S.-H. Lee, C.-H. Yeh, and C. C. J. Kuo, "Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques," *IEEE Signal Processing Magazine,* vol. 23, pp. 79-89, Mar 2006.

[12] B. M. Wildemuth, G. Marchionini, M. Yang, G. Geisler, T. Wilkens, A. Hughes, and R. Gruss, "How fast is too fast? Evaluating fast forward surrogates for digital video," in *Digital Libraries. Proceedings of Joint Conference on*, 2003, pp. 221-230.

[13] K. A. Peker, A. Divakaran, and H. Sun, "Constant pace skimming and temporal sub-sampling of video using motion activity," in *Proceedings of International Conference on Image Processing*, 2001, pp. 414-417.

[14] Z. Gang, L. T. Chia, and Y. Zongkai, "MPEG-21 digital item adaptation by applying perceived motion energy to H.264 video," in *Proceedings of International Conference on Image Processing,* 2004, pp. 2777-2780.

[15] O. A. Lotfallah, M. Reisslein, and S. Panchanathan, "Adaptive-video transmission schemes using MPEG-7 motion intensity descriptor," *IEEE Transactions On Circuits And Systems For Video Technology,* vol. 16, pp. 929-946, August 2006.

[16] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions On Circuits And Systems For Video Technology,* vol. 13, pp. 560-576, July 2003.

[17] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4 advanced video coding standard and its applications," *Communications Magazine, IEEE,* vol. 44, pp. 134-143, Aug. 2006.

[18] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 17, pp. 1103-1120, 2007.

[19] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of H.264/AVC for Multiview Video Compression," in *Image Processing, 2006 IEEE International Conference on*, 2006, pp. 2981-2984.

[20] S. Devillers, C. Timmerer, J. Heuer, and H. Hellwagner, "Bitstream syntax description-based adaptation in streaming and constrained environments," *Multimedia, IEEE Transactions on,* vol. 7, pp. 463-470, June 2005.

[21] D. De Schrijver, W. De Neve, K. De Wolf, S. Notebaert, and R. Van de Walle, "XML-based customization along the scalability axes of H.264/AVC scalable video coding," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, 2006, p. 4pp.

[22] "ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2),Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video," 1994.

[23] "ITU-T Rec. H.263,Video Coding for Low Bit Rate Communication,v3," 2000.

[24] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of Hierarchical B Pictures and MCTF," in *Multimedia and Expo, 2006 IEEE International Conference on*, 2006, pp. 1929-1932.

[25] G. Marchionini, B. M. Wildemuth, and G. Geisler, "The Open Video Digital Library: A Möbius strip of research and practice," *Journal of the American Society for Information Science and Technology,* vol. 57, pp. 1629-1643, 2006.

[26] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: a compressed domain approach," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 14, pp. 462-474, April 2004.

[27] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proceedings of International Conference on Image Processing*, 1998, pp. 866-870.

[28] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using Delaunay clustering," *International Journal of Digital Libraries,* vol. 6, pp. 219-232, 2006.

[29] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, pp. I-511-18.

[30] L. Herranz, "Integrating semantic analysis and scalable video coding for efficient content-based adaptation," *Multimedia Systems,* vol. 13, pp. 103-118, August 2007.

[31] S. Jeannin and A. Divakaran, "MPEG-7 visual motion descriptors," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 11, pp. 720-724, 6 2001.