Integrating semantic analysis and scalable video coding for efficient content-based adaptation*

Luis Herranz

Grupo de Tratamiento de Imágenes, Escuela Politécnica Superior

Universidad Autónoma de Madrid, Madrid, Spain

luis.herranz@uam.es

Abstract Scalable video coding has become a key technology to deploy systems where the adaptation of content to diverse constrained usage environments (such as PDAs, mobile phones and networks) is carried out in a simple and efficient way. Content-based adaptation and summarization are fields that aim for providing improved adaptation to the user, trying to optimize the semantic coverage in the adapted/summarized version. This paper proposes the integration of content analysis with scalable video adaptation paradigm. They must be fitted in such a way that the efficiency of scalable adaptation is not damaged. An integrated framework is proposed for semantic video adaptation, as well as an adaptive skimming scheme that can use the results of semantic analysis. They are described using the MPEG-21 DIA tools to provide the adaptation in a standard framework. Particularly, the case of activity analysis is described to illustrate the integration of semantic analysis in the framework, and its use for online content summarization and adaptation. Overall efficiency is achieved by means of computing activity using compressed domain analysis with several metrics evaluated as measures of activity.

1 Introduction

In the current multimedia scenario, many possible ways of access to the content are available, as different and heterogeneous terminals and networks can be used by the potential users. Adaptation[1] is a key issue to bring the content from the service providers to the actual users, each one using their own terminals and networks, with their own capabilities and constraints, enabling the so called Universal Multimedia Access (UMA)[2]. Especially important is the case of mobile devices, such as PDAs and mobile phones, where other issues as limited computational resources and low power consumption requirements become very important and low-complexity codecs and decoders are necessary. MPEG-21 is a standarization effort to develop a interoperable multimedia framework. This framework is built around the concept of Digital Item (DI). The DI is the basic unit of transaction in MPEG-21, including a standard representation, identification and metadata. Digital Item Adaptation (DIA) is the part of the standard that tackles the problem of adaptation of DI to heterogeneous usage contexts. DIA specifies adaptation related metadata, including those related to the usage environment such as terminal capabilities, network and user characteristics.

Scalable coding is an elegant solution to adaptation[3]. A scalable video stream contains embedded versions of the source content that can be decoded at different resolutions, frame rates and qualities, simply selecting the required parts of the bitstream. Thus, scalable video coding allows a very simple, fast and flexible adaptation framework to a variety of terminals and networks, with different capabilities and characteristics. The numerous advantages of this coding paradigm have motivated an intense research activity with the development of new wavelet based video codecs[4] and the forthcoming MPEG-4 SVC standard[5].

Adaptation techniques, such as scalable video adaptation, transcoding or transmoding, lead to a modified version that usually will result in a reduction of the fidelity of the content. This information loss is distributed uniformly in the whole content, reducing the resolution, the frame rate or the quality. However, this

*Work partially supported by the European Commission under the 6th Framework Program (FP6-001765 - aceMedia Project). This work is also supported by the Ministerio de Ciencia y Tecnología of the Spanish Government under project TIN2004-07860 (MEDUSA) and by the Comunidad de Madrid under project P-TIC-0223-0505 (PROMULTIDIS).
The final publication is available at Springer via http://dx.doi.org/10.1007/s00530-007-0090-0

approach is blind to the content itself, as it does not know anything about what is actually happening. As the final user will prefer to know what is happening in the content, blind adaptation sometimes can lead to some loss or severe degradation of important information for the user. It would be better to try to preserve those parts more “semantically” relevant. Low level semantics extracted from the analysis of the content can help to perform a better adaptation and personalization[6].

There are many applications where such information loss is desirable at some extent. In these applications, the user wants to have a shorter or smaller, but still informative version. A short trailer of the video or a collection of relevant frames can be very helpful for the user to have in mind what happens in the content in less time than playing the whole content, and in a more useful way than reading the title or an abstract. This set of keyframes or segments of video need to be selected from the source content according to some semantic criterion, as they should be representative of as much of the whole content as possible. Semantics enable a better retrieval and browsing of the multimedia content, providing to the user ways to access and browse large multimedia repositories more effectively

To extract the semantics from the content some kind of agent or system needs to analyze the content. One solution is manual annotation but it is very time consuming, hard, expensive and unfeasible for most applications. Automatic analysis is very important to extract semantics avoiding human participation. High level analysis and understanding is still very difficult and complex for automatic systems (the so called semantic gap), but low level analysis is much easier and very helpful for semantic adaptation. Specially, compressed domain analysis gives the efficiency required to allow fast solutions, avoiding full decoding and taking advantage of the compressed domain data as usable features.

The need to effectively browse and manage the huge amount of video content available in personal and commercial repositories, and the possibility of access from diverse terminals and networks, makes essential the use of systems combining summarization techniques and adaptation for effective delivery to the user. This paper contributes exploring the use of semantic analysis in the adaptation process of a scalable video stream, in an integrated framework. It also discusses how the analysis can take advantage of the compressed domain and the scalability, in order to keep efficient the whole adaptation engine. The second contribution is a specific video skimming scheme for scalable video, developed within this framework, and the way that it is described using MPEG-21 DIA tools. In this scheme, the results of semantic analysis guide the temporal adaptation of the content, but keeping the advantages of scalable video adaptation.

The semantic analysis plays an important role in the quality of the resulting summaries/skims, but the semantic analysis itself is not in the scope of the paper, which is focused on the discussion and development of an integrated adaptation framework. The framework is generic enough to use a variety of semantic analysis approaches. However, the framework and the skimming method are illustrated with a simple low level analysis based on computing the activity using compressed domain data.

The rest of the paper is organized as follows. Section 2 briefly discusses the related work. Then, Section 3 describes a generic framework for semantic adaptation of scalable video which is used in Section 4 to propose a scheme for content based skimming. In Section 5, the scheme is integrated in the MPEG-21 DIA adaptation tools. In Section 6 and 7 experimental tests are presented. Section 8 discusses the proposed framework and related approaches, and finally Section concludes the paper.

2 Related work

2.1 Analysis for video adaptation and summarization

Conventional video adaptation (e.g. transcoding) does not take into account that the sequence is actually transmitting information with some meaning for the user. Any level of understanding of the structure or the semantics in the sequence can be useful to take adaptation decisions that improve the quality and meaninfulness of the sequence perceived by the user[1, 6, 7]. Dynamic frame dropping is an example of adaptation technique using information from analysis[8].

Summarization can be considered as a specific type of semantic/structural adaptation where the main objective is the reduction of the information removing semantic redundancy for efficient browsing and presentation of the content. Static storyboards are widely used in video browsing and retrieval to represent the content in few keyframes[9-11]. Several representative frames are selected according to some analysis algorithm in a way that the semantic coverage of the content is optimized and represented in few specific frames.

At a higher semantic level, the sequence can be structured as a collection of shots, in which the relevant frames are selected using motion information or feature clustering. Complex representations have been investigated to model the relationships between shots in more abstract units (scenes), in order to obtain more meaningful summaries for retrieval[[12]. Sometimes the selected frames are not presented as a collection of

images, but in a shorter video sequence, resulting in a fast preview of the content [13].

Video skims are another modality of video abstraction, similar to summaries, but usually built extracting significant segments from the source sequence rather than individual frames. A number of approaches have been used to generate video skims, including visual attention[14], rate-distortion[15], graph modelling[16] and image and audio analysis[17].

Efficient video analysis for adaptation and summarization has been also explored in MPEG compressed domain. The analysis in compressed domain is constrained by the data available in the bitstream, such as DCT coefficients, motion vectors and macroblocks modes in MPEG-1/2 video codecs. Although the analysis is more limited than in uncompressed domain, the main advantage is the high efficiency. Activity and motion features have been widely used[13, 18] as criteria to select or drop frames. Camera motion can be also extracted from the compressed data. The specific coding structure can be also also exploited to drop motion compensated frames[19, 20]. [18] uses MPEG-21 DIA tools to efficient drop frames of a H.264 coded bitstream using simple frame dropping operations.

2.2 Fully scalable video

Scalable coding of media has received an increasing interest in recent times. The advantage of a scalable bitstream is that it can be encoded once, but multiple versions can be decoded, each of them being suitable for different practical usage environments. Besides, the cost of building these adapted versions is extremely low compared with conventional adaptation techniques, such as transcoding, due to the embedded structure of the scalable bitstream.

A fully scalable video bitstream is organized according to a three dimensional spatio-temporal-quality structure where the parts (or atoms) are arranged as blocks in a cube where each coordinate *(s, t, q)* in this space represents an adapted sequence, and determines which blocks should be included in the adapted bitstream (see Figure 1). The increment of the coordinate in one of the dimensions corresponds to an additional layer that enhances the adapted sequence in that dimension.

The scalable bitstream is processed in a GOP (Group of pictures) basis (each GOP is the adaptation unit), and depending on the constraints in the usage environment, the three independent coordinates *(s, t, q)* should be determined, in order to obtain the desired adaptation. Note that several properties of the GOP such as bitrate or PSNR are functions of these three coordinates.

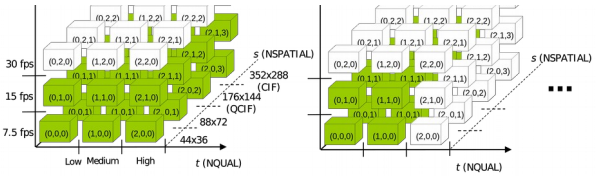


Figure 1. Representation of the first two GOPs of a fully scalable video bitstream.

There are a number of techniques developed to achieve spatial and temporal scalabilities. Interframe wavelet codecs[4, 21, 22] use hierarchical subband decompositions to provide an embedded representation for spatial scalability. Temporal scalability is achieved using a hierarchical subband decomposition of frames within a GOP. This decomposition is combined with motion compensation (Motion Compensated Temporal Filtering-MCTF[23]) to improve the coding efficiency.

The other major category of scalable video codecs are the H264 extensions, most notably MPEG-4 SVC[5]. It uses H264/AVC as base layer and the rest of the spatial enhancement layers are coded exploiting inter-layer prediction. For temporal scalability, hierarchical prediction structures with motion compensation are used.

3 Adaptation framework

This section discusses the design of a generic framework combining efficient scalable video adaptation for applications that need some level of understanding of what happens in the content, in order to improve the utility of the adaptation to the user. It also enables more complex adaptation modalities using this low level of understanding of the content. In order to try to fit these two aspects into an efficient integrated framework, a number of requirements are discussed.

The adaptation engine should be aware of the usage environment. This first requirement is essential, as no adaptation is possible if no information about the context to adapt is available. MPEG-21 DIA[2] tools include the Usage Environment Description (UED) tool to detail the characteristics of networks and terminals. Scalable video coding can use this context information and perform this *context-aware* adaptation in a very efficient way.

The adaptation engine needs a mechanism to add semantic information, in order to guide the adaptation process and enable semantic-based applications. Such *content-aware* adaptation is required for content-based summarization. This semantic information of the content must be available to the adaptation engine, either as previously stored metadata or either obtained from an analysis stage.

A desirable third requirement is *online processing*. This requirement is not always necessary, but it is

imposed to allow applications in scenarios where future content is not available. Online processing refers to the processing of the content as it arrives, with no need of future content (usually because it is not available). Live broadcasting, videoconference and surveillance are examples of scenarios where online processing is useful, but other scenarios can also benefit from the online processing. The main advantage of online processing is the low delay in the delivery of the adapted content. Another advantage is the low processing and storage requirements that usually need these approaches.

However, there is an inherent problem in online processing for video summarization and indexing, and in general, for all the techniques that need a temporal structure of the content into meaningful segments. Online processing is causal, and temporal condensation techniques that try to optimize the coverage using semantics are of necessity non-causal, as they require all the content analyzed in order to build a meaningful summary. Even for human users it is not possible to summarize a movie without having seen it completely. Buffering parts of the content is useful to relax that problem, but there is always a trade off between coverage and delay in the delivery. So this requirement will lead to non optimal summaries, but still can be useful enough to build fast summaries and previews.

Finally, it is always highly desirable to keep the whole framework efficient. Effciency is the main motivation to develop the framework proposed. Usually, most of the applications requiring online processing will also require efficiency, in order to be able to process all the content at it arrives.

Although the importance of semantic analysis is critical for good video abstraction, the paper is not focused on any specific analysis algorithm, but in a flexible framework that could easily include other analysis methods. In order to take advantage of the scalability and the compressed domain, some ideas are discussed to obtain meaningful data directly from the bitstream. These points are shown later with a specific activity based analysis.

3.1 Semantic adaptation engines

An adaptation engine should be able to generate sequences that satisfy a given set of constraints of the usage environment. Then, the terminal will be able to decode the adapted sequence. Using scalable video, the adaptation engine is a simple bitstream extractor that truncates the bitstream selecting only the parts required according to the constraints (see Figure 2a). Thus, the adaptation process is very fast and performed online. In this case, the adaptation is completely blind to the content itself.

A more complex adaptation engine can use the semantics in the content to guide and improve the adaptation process. Content-based summarization and skimming are also possible as part of the adaptation engine. The semantic data could be generated previously and stored as metadata along with the content (see Figure 2b). Standards such as MPEG-7 provide standard description tools for specifying low level semantics that can be used in summarization[24]). In this architecture, the adaptation process is driven by metadata[25], and the computational burden due to the analysis is avoided at time of adaptation and shifted to a previous stage. There are no constraints in the efficiency and complexity of the analysis algorithms. It is suitable for scenarios where the content is created and stored before the consumption, such as retrieval of video from previously stored content repositories. The adaptation is still very efficient and online, with knowledge of the semantics of the whole content. When content metadata is available it should be used as it can help to improve the adaptation and summarization without a significant overload of the system.

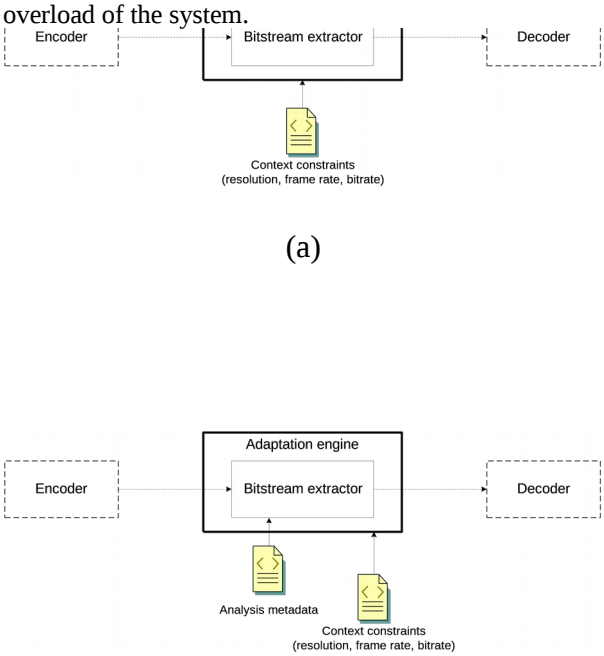


Figure 2. Adaptation engines: (a) Content independent; (b) Content aware (metadata-driven)

However, metadata is not always available. Besides, in many scenarios the content should be delivered and adapted online just after its creation with low delay, and the analysis and adaptation are very constrained. In this case, the analysis should be included in the adaptation engine. The architecture of such content based adaptation engine includes an analysis stage that will guide the bitstream extractor (see Figure 3).

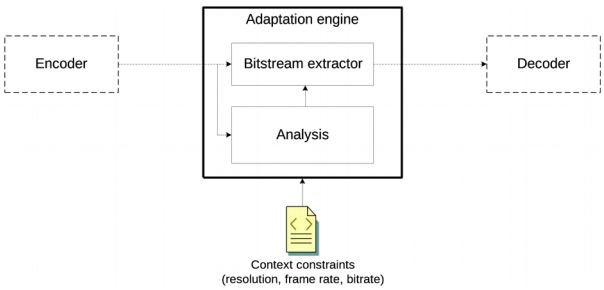


Figure 3. Proposed content aware adaptation engine, with online analysis.

Note that if the engine is required to process online the content, the analysis will be required to be online too. Thus, most of the semantic summarization algorithms cannot be used, as they need to have all or at least certain amount of the content preprocessed (shot, or scene) to extract a summary.

In this last architecture, it is important to be careful, because one of the key advantages of scalability is that adaptation engines are extremely efficient. The semantic adaptation should be also very efficient in order to keep the advantages of the scalable video framework. To satisfy this efficiency requirement, the analysis should be very efficient as well. For this reason, it is very desirable that the analysis stage work directly over the compressed domain parameters present in the coded bitstream, in order to avoid decoding. The rest of the paper is focused on this architecture using compressed domain analysis to provide efficient analysis and keeping the whole adaptation process efficient.

3.2 Compressed domain analysis in scalable video

Working directly on the compressed domain for efficient analysis has been extensively explored for coding standards like MPEG-1/2/4, based on hybrid video coding using spatial DCT together with motion compensation. Compressed domain analysis avoids most of the decoding stages, but more important, dramatically reduces the amount of data to process(e.g. DC images[26]) and provides meaningful parameters that can be used to extract useful features for analysis[27]. There is a wide variety of experimental fully scalable codecs, most of them sharing common approaches to the scalability that can be used to extract useful parameters. There are two features that can be extracted easily from most of these codecs in a generic way:

- Low resolution images. Discarding all the spatial enhancement layers. Only the base layer is required to be decoded. Discarding temporal layers can further reduce the decoding time, as no inverse temporal steps are required. These low resolution frames carry almost the same information as the full resolution source frames, and can be used as input of many analysis algorithms without significant degradation of the results.
- Motion fields. Temporal interframe redundancy is usually exploited using a motion compensated approach, either MCTF, or either hierarchical prediction structures. In both approaches there are different sets of motion vectors coded in headers, giving a coarse representation of the motion of the objects in the frames.

These basic features can be used by the analysis algorithms to obtain higher level semantics, such as shot change detection[28], activity analysis[29], camera motion[30] and coarse spatial segmentation[31], among others, in the same way as they are widely used in MPEG-1/2 working with motion vectors and DC images. Section 6 describes some experiments where compressed domain data from a scalable video codec are used to perform activity analysis, using the basic features discussed previously.

4 Video skimming of scalable video

Previous section has discussed the abstract model to provide semantic adaptation integrated with scalable video adaptation. This section and the rest of the paper focuses on a specific adaptation scheme that uses the previous model, taking advantage of the temporal scalability to provide semantic selection of frames (skimming). In this scheme, the adaptation engine can vary the frame rate of the adapted sequence depending also on the semantics of the content.

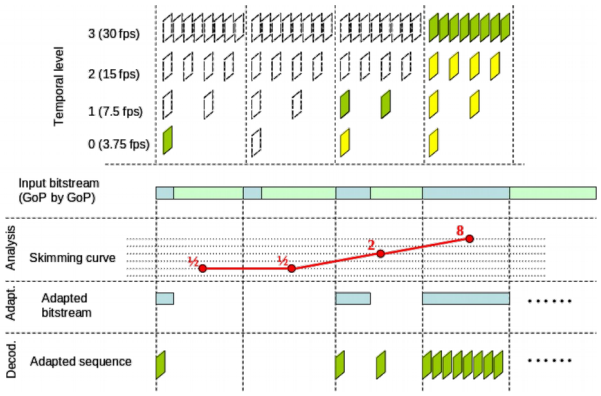


Figure 4. Proposed scheme for content based skimming of scalable video.

The skimming scheme is depicted in Figure 4. The semantic information is linked to the bitstream extraction using a skimming curve, indicating the number of frames that should be selected for each GOP. This skimming curve is the result of the analysis stage. Due to temporal scalability, each GOP will have several possible temporal versions. Depending on the value of the skimming curve, the most suitable temporal version

of the GOP will be selected from the input bitstream, and included in the adapted bitstream.

The skimming curve can be obtained by means of semantic analysis of the content or from previously stored metadata (following the metadata-driven approach of Figure 2). The quality of the summary/skim will depend on the capability of the semantic analysis to extract the relevant frames of the sequence for a given application (e.g. skimming, summarization, fast browsing) and context. However, complex analysis algorithms are out of the scope of this paper, instead a simple method is described to illustrate the use of low level semantics in the framework. An extended approach is the use of an activity or motion intensity index that can easily be converted to a skimming curve, and also can be extracted online.

4.1 Activity based skimming

Activity has been proposed as a measure of summarizability of a video sequence[12], where static parts of the video sequence can be represented with few frames, and high activity parts due to motion need more frames to cover the events of the frames. Following this reasoning, activity analysis is used to obtain the skimming curve. An activity measure is given to each GOP in the source sequence using information present in the coded bitstream. Using the skimming scheme proposed (see Figure 4), the number of frames available for each GOP will depend on the scalability framework, and will be selected according to its activity.

Let T be the number of temporal decompositions (there will be T+1 levels, from 0-lowest frame rate- to T-highest frame rate-). Assuming that level *0* has 1 frame per GOP the length of the GOP will be

$$L = 2^T \quad (0)$$

and the number of frames per GOP at level t

$$L^{t)} = 2^t \quad t \in \{0,1,2,\cdots,T\} \quad (0)$$

In the scheme, the activity is assumed to vary linearly with the measured activity. The skimming curve r_k is then related with the activity a_k of the GOP k with a constant K:

$$r_k = K \cdot a_k \quad (0)$$

Due to the dyadic structure used in temporal scalability, each additional temporal layer doubles the number of frames in the GOP. If M is the number of desired skimming levels in the skimmed sequence, the possible number of frames that the GOPs of the adapted sequence will have is in the set

$$SK = \{2^{T-M+1},\cdots,2^T\} \quad M > 0$$

(0)

The temporal level to be selected will be the result of quantizing the skimming curve r_k in the SK set, and gives the number of frames per GOP in the skimmed sequence

$$m_k = \lfloor r_k \rfloor_{\in SK} \quad (0)$$

The previous skimming curve could be used to guide the bitstream extraction process. However, sudden variations in the curve lead to too many fast and continuous accelerations and disaccelerations that are undesirable in the skimmed sequence. In order to keep a more natural behaviour in the skimming, a simple smoothing filter has been used.

$$m'_k = \lfloor 0.7\, r_k + 0.3\, r_{k-1} \rfloor_{\in SK} \quad (0)$$

It is convenient to use parameters directly related with the temporal scalability in terms or temporal levels more than frames per GOP. The parts that will remain in the bitstream are those required to decode each GOP with at a given number of frames m'_k, corresponding to the temporal level:

$$p_k = \log_2 m'_k \quad (0)$$

Note that M can be higher than the number of available temporal levels, and the number of frames per GOP can be lower than 1 frame. (0) is only valid when m'_k is 1 frame per GOP or greater. When the number of frames per GOP is below 1 frame, only is kept the lowest frame rate version (temporal level 0) of the GOP, and a number of the subsequent GOPs will be skipped, according to the values of the skimming curve, in order to achieve the desired frame rate. This skipping procedure is signalled to the adaptation stage with an extra variable $skip_k$ that is true when the GOP is skipped.

The bitstream extractor will use the values of p_k and $skip_k$ to achieve the desired temporal adaptation. Note that, regarding to spatial and quality adaptation, the behaviour of the adaptation engine is non content-based, adapting to the constraints specified in the usage environment. Only those parts required for decoding the given frame size and quality will be kept in the adapted bitstream.

4.2 Application to dynamic frame dropping and fast browsing

This scheme enables the use of the scalable video adaptation for applications using content based skimming of frames.

One of such applications is fast browsing[13] of video sequences. The input sequence is compacted in an adapted sequence with fewer frames. Played at full frame rate (or the highest frame rate allowed by the terminal), the adapted sequence can be much shorter, but still with most of the coverage of the content. It can be useful when the user wants a fast glance of the video content (a quick video summary), but trying to keep the semantics in terms of 'pace' in the sequence, skipping faster low activity parts and focusing on those parts where more activity is detected. In the case of using the proposed adaptation scheme for fast browsing, the adapted sequence is played with constant frame rate, but with variable GOP rate, as the number of frames of each GOP is selected dynamically.

If the same set of selected frames is played at *constant GOP rate*, instead of playing at constant frame rate, the length of the adapted sequence will be the same. The adapted sequence will be presented in the terminal as the same sequence but with fewer frames in static parts and more frames in more active ones. Such adaptation can be seen as an adaptation engine using dynamic frame dropping[20] in scalable video.

5 Adaptation to the usage environment

At this point, the usage environment has not being considered yet. The problem of adapting bitstreams to a given usage context can be formulated using some tools of MPEG-21 DIA, designed to describe generic optimization problems and the constraints involved in the adaptation process[32].

The decision problem in DIA is formulated using the description tools Universal Constraints Description (UCD) and Adaptation Quality of Service (AQoS). The UCD tool specifies directly the constraints involved in the problem (e.g. maximum width, maximum bitrate) using numeric expressions that relate the constraining value with the variable that is being constrained. The AQoS tool specifies the dependencies between the variables (termed as IOPins in MPEG-21 DIA) involved in the optimization problem. Some of these variables are independent, while the rest depend on the other variables. The decision-taking engine processes the input bitstream selecting the best adaptation variables using the information available in AQoS and UCD for a given adaptation unit. Then, the bitstream adaptation engine performs the actual extraction and modification of the bitstream according to the adaptation variables.

A third adaptation tool that is usually used together with UCD and AQoS is the UED, where a description of the usage context can be described. As the objective is the adaptation to a given constrained usage context, most of the constraints specified in UCD are related to the capabilities described in UED (e.g. terminal size, network bitrate).

5.1 Scalable video adaptation in MPEG-21 DIA

In this work, the MPEG-21 DIA approach is followed, using the same tools and concepts as in the case of conventional scalable video adaptation, and extending it to include the semantic adaptation as new elements in the optimization problem. The problem of adapting fully scalable video consists of selecting the adaptation coordinates (*s*, *t*, *q*) of each GOP (the adaptation unit) that gives the best adapted version for a given set of constraints.

In the case of fully scalable video, usually the optimization problem is to maximize a measure of the quality, given the constraints of the usage environment (network and terminal). In this application, the metric selected to be maximized is an ad-hoc metric (PSNRGOP) based on the peak signal to noise ratio (PSNR) from the mean square error of the full GOP of the frames upsampled to the highest frame rate. For a GOP with spatial level *s*, temporal level *t* and quality level *q* is computed as

$$PSNRGOP_k^{s,t,q)} = 10\log_{10}\left[\frac{255^2}{MSE\left(STU^{S,Q)}\left(g_k^{s,t,q)}\right), g_k\right)}\right]$$

where MSE is the mean squared error, $STU^{S,Q)}$ is the spatiotemporal upsampling operator to the highest temporal and spatial levels, and $g_k^{s,t,q)}$ is the reconstructed GOP. Figure 5 shows the PSNRGOP computed for different temporal levels and different quality layers. However, a proper objective perceptual metric is desirable in order to measure better the real

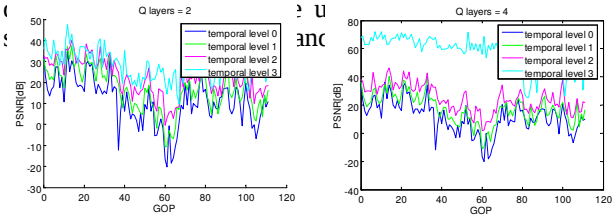


Figure 5. Quality measure PSNRGOP for 2 and 4 temporal layers.

The optimization problem involved in this application can be expressed as:

maximize PSNRGOP=f(NTEMP, NSPATIAL, NQUAL; GOP)

subject to

FRAMEWIDTH ≤ *display_width*
FRAMEHEIGHT ≤ *display_height*
FRAMERATE ≤ *display_refresh_rate*
BITRATE ≤ *max_bitrate*

The constraints are imposed by the usage environment (described in the UED tool). The independent variables to be optimized are NSPATIAL, NTEMP and NQUAL (equivalent to the *(s,t,q)* adaptation coordinates), indicating the number of spatial, temporal and quality layers in the adapted GOP. The rest of variables are not independent, and can be obtained as functions of NSPATIAL, NTEMP and NQUAL, and the number of the GOP, declared in the AQoS tool.

5.2 Adding semantic constraints

The scheme of activity based skimming of scalable video described previously can be included in the MPEG-21 DIA framework modifying the optimization problem in order to include the results of analysis as new constraints and variables. The skimming level p_k limits the temporal coordinate *t* (represented by NTEMP). The GOP skipping can be included as a new variable INCLUDE_GOP signaling if the GOP should be included or not according to the value of $skip_k$. The new optimization problem with these semantic constraints is:

*maximize PSNRGOP=f(NTEMP, NSPATIAL, NQUAL, **INCLUDE_GOP**; GOP)*

subject to

FRAMEWIDTH ≤ *display_width*
FRAMEHEIGHT ≤ *display_height*
FRAMERATE ≤ *display_refresh_rate*
BITRATE ≤ *max_bitrate*
NTEMP ≤ (*analysis_temporal_level* = m_k)
INCLUDEGOP = NOT($skip_k$)

These two new constraints are imposed by the activity analysis stage in the previously described framework. Note that bitrate should take into account, if necessary, the fact of the speeding up of the sequence and consequently the GOPs as adaptation units. When the presentation of the adapted sequence includes time reduction (e.g. fast browsing), each GOP can have different number of frames but presented in a constant frame rate. Thus, GOPs with fewer frames should be delivered faster than those with more frames.

The value of *analysis_temporal_level* can also be computed previously and stored along with the content and the rest of adaptation metadata, usually in the same AQoS descriptor, in a fully metadata-driven architecture (see Figure 2b). However, this work addresses the architecture guided by compressed domain analysis, where the activity is computed online as the content arrives (see Figure 3). A hybrid architecture with caching can take advantage of the compressed domain analysis the first time that the content is requested and store the measure of activity or the temporal level in an additional AQoS descriptor, and then follow a fully metadata-driven approach for the next adaptation.

5.3 Personalization aspects

Each user could have different preferences on the amount of skimming for fast browsing. In the proposed system, it depends on the minimum activity threshold *K* and the number of levels *M* in the skimming curve, and it will lead to different skimmed versions. These two parameters could be specified by the user or predefined in a personalization profile. However, they cannot be included directly in as preferences in the summarization.

A more intuitive parameter could be the minimum skimming ratio, expressed as

$$ratio = \frac{FRAMES\ IN\ SUMMARY}{FRAMES\ IN\ SEQUENCE} = 2^{-(M-1)}$$

where the number of levels *M* can be easily derived. As at least 1 frame each 2^{M-1} frames, this ratio allows the user to express a minimum, in the case of static sequences. In practice, more frames are selected, as usually the videos have activity, and the smoothing filter will also affect the final number of frames per GOP.

From the user's point of view, it is more useful to quantize the parameter K in some levels and include it in the profile in terms of predefined values linked to high, medium or low amount of skimming.

6 Experiments on Activity Analysis

The activity measure should reflect as much as possible the actual activity present in the content. The performance of the adaptation process will rely on it. The activity should be computed in a GOP basis, as it is the basic unit in the adaptation engine.

However, in order to keep an efficient adaptation engine even when there is no previous activity data, the measure should be rapidly computed from the compressed domain parameters present in the bitstream, avoiding decoding or partial decoding as much as possible.

6.1 Data extraction for analysis

As it was pointed previously, the extraction of useful features from the coded bitstream is of necessity dependent on the codec itself, its format and implementation. In this section we will focus on wavelet based video coding and the specific implementation used for this paper. Wavelet coding enables a natural multiresolution framework for highly scalable video coding. Most works on this subject are based on two transforms: a wavelet transform in the temporal axis, combined with motion compensation (MCTF), and another 2D spatial discrete wavelet transform (2D DWT). There are many codecs that use the framework, known as t+2D framework, where the temporal transform is performed before the spatial transform. In this paper we consider the specific codec described in [22, 33]. It uses MCTF with hierarchical variable size block matching (HVSBM)[34] and forward motion compensation.

To extract low resolution images we can take advantage of the spatial scalability to avoid almost all the decoding. In the coded bitstream, the lowest frame size version embedded in the stream is used. Discarding all the spatial subbands except the lowest one, the amount of data is highly reduced, all the inverse 2D DWT are avoided and inverse MCTF is performed over a number of pixels much lower. Thus, the decoding time is dramatically reduced, as most of the decoding is not necessary. In the case of the studies presented in this section, only a frame per GOP is required. We can take advantage of the temporal scalability to further reduce the decoding steps, noting that the first temporal layer already is the low pass temporal subband containing the small resolution image, so there is no need of inverse MCTF (see Figure 6). Only texture or entropy decoding is required.

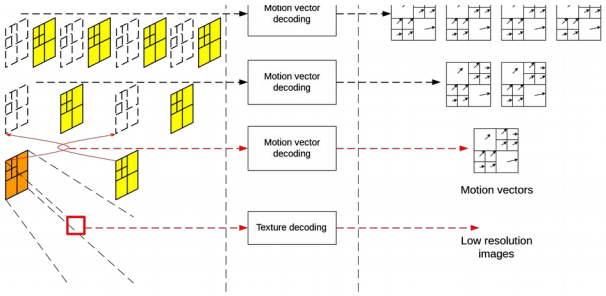


Figure 6. Extraction of compressed domain data for analysis from a wavelet scalable video codec.

Due to the hierarchical motion structure of the dyadic decomposition using MCTF, different sets of motion vectors for each temporal level are available in the GOP. Depending on the analysis algorithm, one set can be more useful than others. Motion vectors are usually coded in the header and can be easily extracted without further decoding. However this feature is more codec dependent, as it also depends on the block matching algorithm used.

6.2 Motion Activity

Motion activity is defined as a measure of the "intensity of motion" or "pace" of a sequence as it is perceived by a human. MPEG-7[13] includes a descriptor of the intensity of motion activity. It is based on the standard deviation of the motion vectors quantized in five levels. For this work, the quantization step is not used, in order to have a continuous motion activity measure. For the case of a scalable video codec, previous experiments[35] showed that it is enough to use only the set of motion vectors from the lower temporal level to estimate the motion activity of a GOP. Using this set of motion vectors (see Figure 6) from the lowest temporal level and including the area of the blocks, the intensity of motion activity descriptor is then computed as:

$$a_k^{MV} = \sqrt{\frac{1}{A}\sum_{i=0}^{N-1} a_k(i)\left\|\vec{m}_k(i) - \vec{\overline{m}}_k\right\|^2}$$

(0)

where N is the number of motion compensation blocks and $\boldsymbol{m(i)}$ is the motion vector at block i. The parameters $a(i)$, A and $\overline{\boldsymbol{m}}\boldsymbol{(i)}$ are respectively the area of the block i, the total area, both in pixels, and the mean motion vector. The use of the area is to cope with veriable block size motion compensation. Figure 7 shows the motion activity computed using the expression as was proposed in (0).

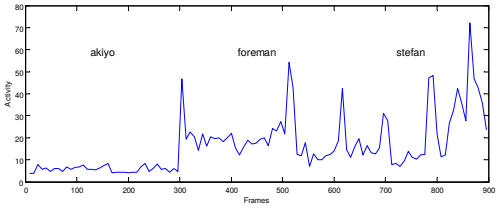


Figure 7. Activity computed using the motion vectors.

An important drawback of the use of motion vectors to perform compressed domain analysis is the presence of a large amount of spurious vectors, which do not represent the real motion in the scene, and will depend on the specific implementation of the encoder and its configuration (usually, search algorithm and window).

6.3 PCA Space Euclidean Metric

Li et al[15] argue that motion activity is still very expensive to compute, and a metric based on Principal

Component Analysis (PCA) is proposed. Each frame is downsampled to a very low frame size WxH, such as 8x6, and projected through PCA to a linear subspace preserving most information in a reduced dimension. The activity is computed as the weighted euclidean distance between the first frames of two consecutive GOPs. Only the six most significant of the 48 coefficients are considered in our experiments:

$$Y_k^{PCA} = T_{PCA}(S_{8x6}(F_k)) \qquad (0)$$

$$a_k^{PCA} = \sqrt{\sum_{i=0}^{5} \left(\hat{Y}_k^{PCA}(i) - \hat{Y}_{k+1}^{PCA}(i)\right)^2 \omega_i} \qquad (0)$$

In (0) and (0), T_{PCA} denotes the PCA transform, S_{8x6} is the scaling operation to 8x6 size arranged in a vector of 48 values, Y_k^{PCA} and $\hat{Y}_k^{PCA}$ are the PCA feature vector of the frame F_k and the truncated feature vector to the first 6 values, where I_{6x48} is the identity matrix truncated in rows to 6x48, ω_i are the weights, and a_k^{PCA} denotes the activity curve in the GOP *k*.

Thus, the activity of the GOP can be computed very fast with this technique, avoiding decoding and using a very low amount of values. In our experiments, the PCA subspace is computed for the whole sequence, so it has the drawback of having to process all the sequence first to obtain the basis, and then process it again to obtain the features. With this approach, it is not possible to adapt on-line the sequence, or avoiding buffering of blocks of GOPs.

6.4 DCT Space Euclidean Metric

A possible variation of the previous metric is the use of the Discrete Cosine Transform (DCT). In this case, the projection subspace is suboptimal, though very close to the PCA subspace when the source is a Markov process of 1st order[36]. In this case, the feature vector is:

$$Y_k^{DCT} = T_{DCT}(S_{8x6}(F_k)) \qquad (0)$$

where T_{DCT} is the 1-D DCT transform. The corresponding activity will be

$$a_k^{DCT} = \sqrt{\sum_{i=0}^{5} \left(\hat{Y}_k^{DCT}(i) - \hat{Y}_{k+1}^{DCT}(i)\right)^2 \omega_i} \qquad (0)$$

The activity curves obtained using the DCT-based metric are very close to those obtained with the PCA-based one (see Figure 8). Besides, DCT can be computed faster and the basis does not need to be computed for each sequence, so it could be used in online adaptation.

Figure 8 also shows that shot changes have an extraordinary high value of activity. Another advantage of using these metrics in a GOP basis, is the possibility of the detection of shot changes by hard thresholding, with a GOP precision, which is enough in many cases. It can be detected both cuts and gradual transitions with lengths up to a half GOP (in the experiments GOP is 8 frames, so up to 4 frames long). This is not possible with the motion activity curve, as there are no motion vectors relating adjacent GOPs.

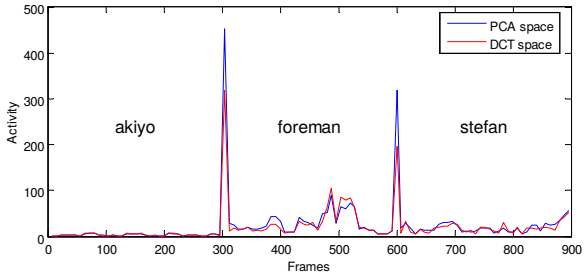
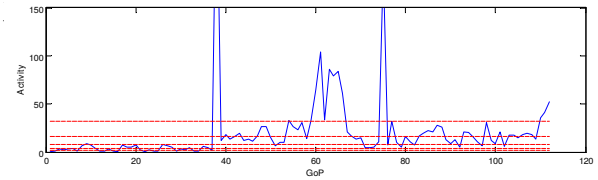


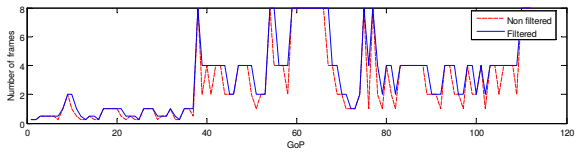
Figure 8. Activity computed in PCA and DCT spaces.

7 Experimental Results

The proposed framework and skimming algorithm were tested with a sequence built chaining three standard sequences: *akiyo* (low activity), *foreman* and *stefan* (medium-high activity). Each of these sequences has 300 frames with CIF resolution at 30 frames per second. This sequence was encoded with the QMUL's wavelet SVC codec][22], using 3 temporal decompositions (4 possible reconstructed frame rates), 3 spatial decompositions (4 possible reconstructed frame sizes) and 4 quality levels, with a GOP of 8 frames. The sequence was adapted to a terminal with a display size of 176x144 and 30 frames per second using the proposed semantic adaptation engine. The only semantic feature used to guide the frame selection in this test is



(a)



(b)

Figure 9. Adaptation of the test sequence. (a) Activity and thresholds; (b) Number of frames per GOP in the adapted sequence before and after smoothing filter.

The metric used to measure the activity was the DCT based. Error: Reference source not founda shows the activity curve and the thresholds used, setting the value of the minimum activity threshold *K* to 0.6, and the number of skimming levels *M* to 8. In Error: Reference source not foundb it is shown the number of frames per GOP before and after the smoothing filter. Comparing both adapted sequences generated from the non filtered and filtered curves, the second has much less variations and looks more natural. The adapted sequence is shown in Figure 10. The frames have been selected according to the activity. A number of 99 frames were selected out of 900. A lot of significance is given to the panning in *foreman* sequence, because the DCT measure (as PCA does as well) gives a high activity measure to the parts with camera motion. If these camera effects are not desirable in the summarized sequence, or desirable to be represented with fewer frames, a proper metric focusing more on object motion should be used.

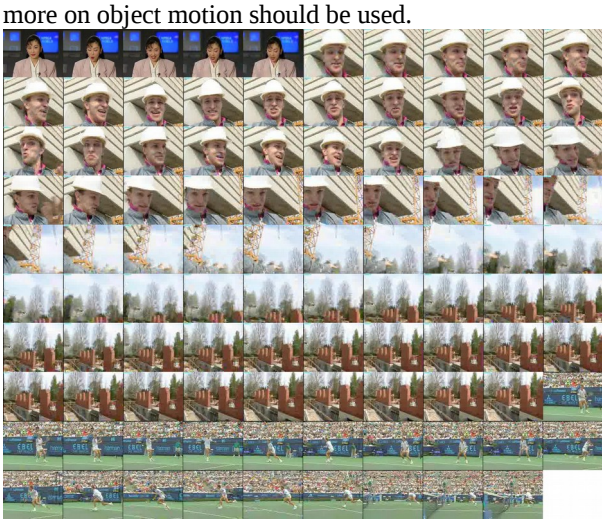


Figure 10. Example of adapted sequence.

Another consequence of discarding temporal layers is the reduction of the decoding time in the terminal, as several MCTF steps are avoided (see Figure 11). This fact can be very useful to reduce the complexity at the terminal side. For the implementation used in the experiments, the decoding time depends more heavily on the quality level, than on the temporal level, due to entropy decoding.

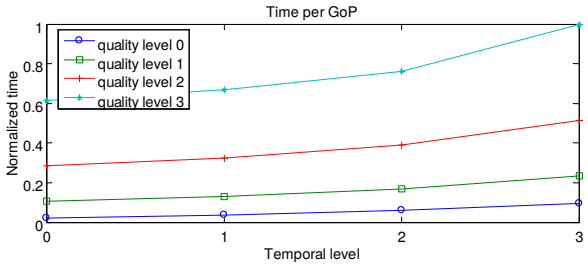
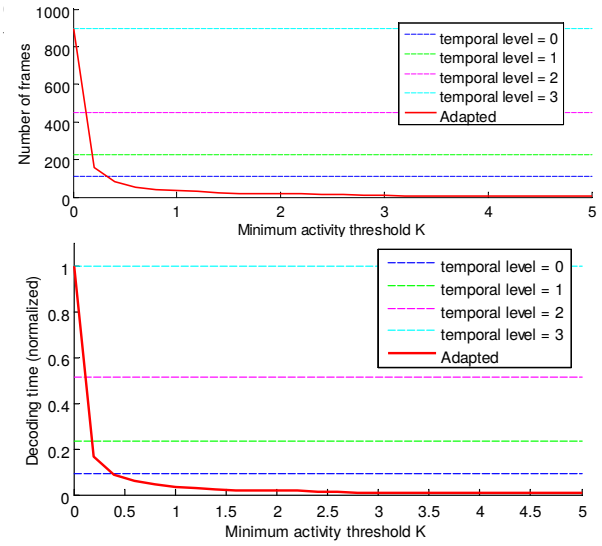


Figure 11. Decoding time (normalized) for full spatial resolution.

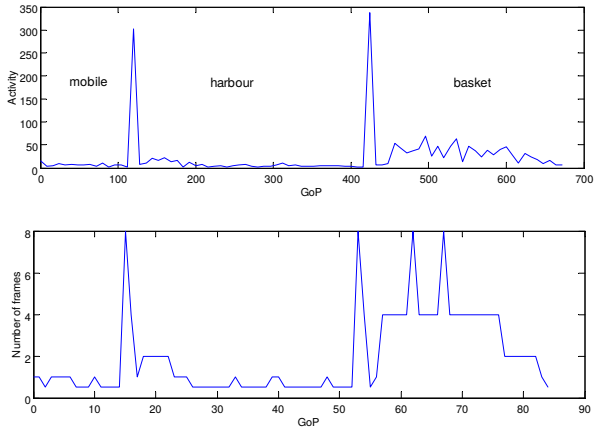
Figure 12a shows the results for different values of the parameters *K* in the test sequence. A higher value will lead to shorter summaries. This summary can be played much faster giving an effective fast overview of the content. If this result is used in adaptation with content based frame dropping the decoding time is highly reduced (see Figure 12b). Note that skipping a GOP takes almost no time. This second application can be very useful to adapt to complexity constrained environment, due to reduced processing capability, or



(b)

Figure 12. Results of adaptation for diffent values of K. (a) Number of frames in the adapted sequence; (b) Decoding time reduction.

A second test was done using a higher resolution sequence (4CIF, i.e. 704x576), built with other three standard sequences: *mobile* (low activity), *harbour* (low-medium activity) and *basket* (high activity). The resulting curve of activity and the number of frames per GoP are shown in Figure 13, for the same set of thresholds as in the previous test. As it was expected, *basket* has a higher number of frames in the adapted sequence (see Figure 14) than *mobile* and *harbour*.



(b)

Figure 13. Results for the second test sequence. (a) Activity; (b) Number of frames per GOP in the adapted sequence.

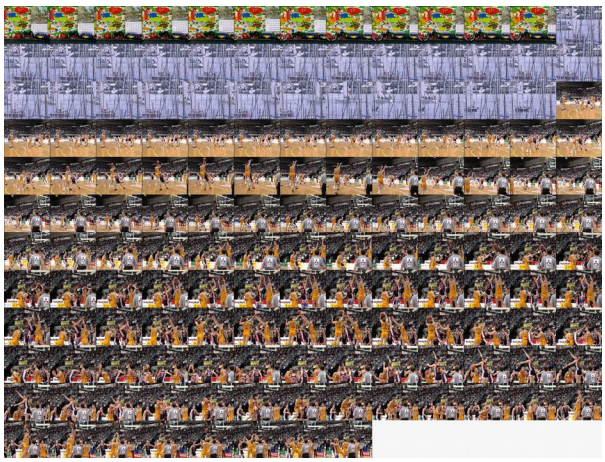


Figure 14: Example of adapted skim for the second test sequence

Finally, in order to show the differences due to the content semantics (activity), the algorithm was tested with the six basic sequences used in the previous test sequences. Figure 15 shows the number of frames selected (in percent) for each of the sequences. Due to the online processing, the number of frames in the adapted sequence depends on the activity of the content, and it cannot be determined prior to the adaptation itself. For a given threshold, more frames are selected in sequences with more activity (*foreman*, *basket*) while few frames are selected in static ones (*akiyo*).

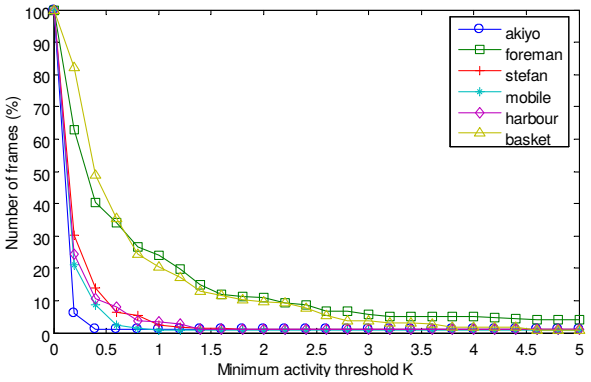


Figure 15. Comparison of the number of frames selected.

8 Comparison and relation with other approaches

The proposed framework and skimming scheme integrate two of the stages necessary to deliver an adapted skim or summary to a user in a constrained environment. Existing approaches focus either on summarization[14-17] or either in adaptation[1, 3, 37]. In summarization works, adaptation usually is not considered and, if necessary, it is added as an additional independent stage, leading to a much less efficient system.

From the adaptation point of view, scalable video coding and transcoding are the main technologies for content-blind adaptation. Transcoding[37] is the most used approach for adaptation of video summaries, usually only encoding from uncompressed frames to video. Compressed domain transcoding helps to improve the efficiency. However, scalable video coding is the most efficient approach to adaptation of the video signal, in contrast to transcoding. Scalable video can be used to extract non content based skims, just selecting a lower temporal version (see Figure 2a). Compared with the proposed approach, it has the drawback of not using the semantic relevance, leading to less useful summaries. Figure 16 shows the skim resulting from the selection of the second temporal level of the bitstream of the second test sequence. Activity is not considered in the adaptation. Although it has a similar number of frames compared to that resulting from the proposed system (see Figure 14), static and very active parts have the same proportion of frames.

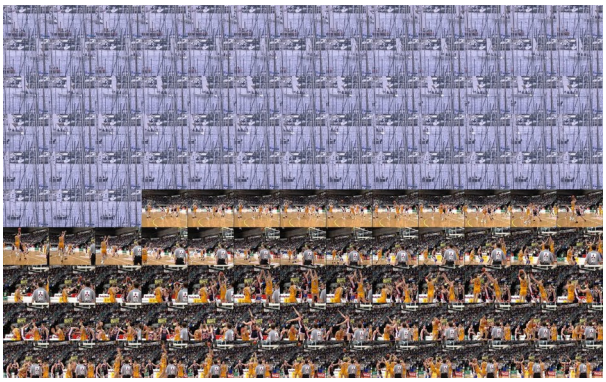


Figure 16: Example of skim generated using a non content based approach

Some approaches for non scalable video adaptation use low level semantics to help the adaptation. [13] uses motion activity for adaptive fast playback of video sequences. [18] uses a similar method to drop frames dynamically in H.264, according to a measure of the perceived motion. In this case, B frames are dropped in parts with low perceived motion. The proposed method takes advantage of the temporal scalability to obtain a similar behaviour.

Metadata-driven approaches[25] to semantic adaptation separate the analysis from the adaptation. They have the advantage of enabling semantic adaptation, as semantic metadata is available before the adaptation, but they need to have parsed and analyzed previously the content (sometimes manually). [25] describes the use of metadata in image transcoding. Figure 2b is the equivalent version of the metadata driven approach for scalable video. The proposed skimming system follows this architecture if the skimming curve is stored previously as metadata and included as semantic constraints as shown in Section 7. It has the advantage of using scalable video and MPEG-21 DIA, so the adaptation is very efficient.

From the semantic point of view, it is difficult to compare the system with other works, as the role of the analysis method in the paper is only illustrative of how semantic analysis can be included into the framework and in the skimming method. The activity analysis is an approach widely used in the literature[13, 18], and the results of the system are comparable to those using activity analysis. It has the drawback of being online (causal) and not optimum, as no future content data is considered and it is not expected to obtain better results than those using sophisticated and high level semantic analysis[14, 16], usually with offline analysis, However the framework is flexible enough to use most of these algorithms, so the semantic quality of the summaries or skims would improve. Besides the proposed method has the advantage of high efficiency and the adaptation integrated with the summarization.

9 Conclusions and future work

This paper discusses the use of semantic analysis with scalable video to enable content based applications using the particularities of the adaptation model, focusing on the efficiency as the main objective. The proposed solution is a framework combining both compressed domain analysis and bitstream extraction. A temporal adaptation scheme has been also proposed, taking advantage of the organization of the temporal layers in the bitstream, and using a simple activity analysis to illustrate the integration of semantic analysis.

The use of the system for fast browsing and dynamic frame dropping of video sequences, with activity as semantic clue, has been studied in the MPEG-21 DIA standard framework, including new semantic constraints into the optimization problem of adaptation. The result is a highly efficient approach, even in the analysis, as only few data extracted directly from the compressed domain are necessary to compute an activity measure.

Experimental results show that, even using a simple activity-based keyframe selection algorithm, the adapted video sequences are useful for applications such as fast browsing of video search results, and are also useful to reduce the decoding requirements in complexity constrained environments, as mobile devices.

However, many aspects can be improved. A key issue to achieve a good summary is the skimming algorithm itself and future work will focus on improving the skimming algorithm including more and better semantic clues, and hopefully achieving a more meaningful summary. Another issue that is expected to be improved is the metric that measures the perceived quality, in order to improve the overall user's experience in the adaptation.

10 Acknowledgments

The author would like to acknowledge Nikola Šprljan, Marta Mrak and Ebroul Izquierdo for their help and support with their scalable video codec, and José M. Martínez for his suggestions and comments.

11 References

1. S.-F. Chang and A. Vetro: Video adaptation: concepts, technologies, and open issues, Proceedings of the IEEE, **93** (1), 148-158 (2005)
2. A. Vetro: MPEG-21 digital item adaptation: Enabling universal multimedia access, IEEE Multimedia, **11** (1), 84-87 (2004)
3. J. R. Ohm: Advances in scalable video coding, Proceedings of the IEEE, **93** (1), 42-56 (2005)

4. J. R. Ohm, M. van der Schaar, and J. W. Woods: Interframe wavelet coding motion picture representation for universal scalability, Signal Processing: Image Communication, **19** (9), 877-908 (2004)
5. H. Schwarz, D. Marpe, and T. Wiegand: Overview of the scalable H.264/MPEG4-AVC extension. In: Image Processing. Proceedings of International Conference on (2006)
6. F. Pereira, P. Van Beek, A. C. Kot, and J. Ostermann: Special issue on analysis and understanding for video adaptation, IEEE Transactions on Circuits and Systems for Video Technology, **15** (10), 1197-1199 (2005)
7. N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor: Applications of video-content analysis and retrieval, Multimedia, IEEE, **9** (3), 42--55 (2002)
8. M. Furini and V. Ghini: A video frame dropping mechanism based on audio perception. In: IEEE Global Telecommunications Conference Workshops, 2004, pp. 211-216 (2004)
9. 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, **7** (5), 771-785 (1997)
10. 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, **9** (8), 1269-1279 (1999)
11. S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg: Abstracting digital movies automatically, Journal Of Visual Communication And Image Representation, **7** (4), 345--353 (1996)
12. 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, **7** (4), 648-666 (2005)
13. K. A. Peker, A. Divakaran, and H. Sun: Constant pace skimming and temporal sub-sampling of video using motion activity. In: Image Processing. Proceedings of International Conference on, pp. 414-417 (2001)
14. 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, **7** (5), 907-919 (2005)
15. Z. Li, G. M. Schuster, A. K. Katsaggelos, and B. Gandhi: Rate-distortion optimal video summary generation, IEEE Transactions on Image Processing, **14** (10), 1550-1560 (2005)
16. 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, pp. 104-109 (2003)
17. 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, 1998. Proceedings., 1998 IEEE International Workshop on, pp. 61-70 (1998)
18. Z. Gang, L. T. Chia, and Y. Zongkai: MPEG-21 digital item adaptation by applying perceived motion energy to H.264 video. In: Image Processing, 2004. International Conference on, pp. 2777-2780 (2004)
19. W. Lai, X. D. Gu, R. H. Wang, L. R. Dai, and H. J. Zhang: Perceptual video streaming by adaptive spatial-temporal scalability, In: Advances in Multimedia Information Processing - PCM 2004, Lecture Notes in Computer Science (3332), Springer-Verlag Berlin, pp. 431-438 (2004)
20. H. J. Cha, J. H. Oh, and R. Ha: Dynamic frame dropping for bandwidth control in MPEG streaming system, Multimedia Tools and Applications, **19** (2), 155-178 (2003)
21. S. T. Hsiang and J. W. Woods: Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank, Signal Processing: Image Communication, **16** (8), 705-724 (2001)
22. N. Sprljan, M. Mrak, G. C. K. Abhayaratne, and E. Izquierdo: A scalable coding framework for efficient video adaptation. In: Proc. Int. Work. on Image Analysis for Multimedia Interactive Services (WIAMIS) (2005)
23. J. R. Ohm: Three-dimensional subband coding with motion compensation, Image Processing, IEEE Transactions on, **3** (5), 559--571 (1994)
24. P. M. Fonseca and F. Pereira: Automatic video summarization based on MPEG-7 descriptions, Signal Processing: Image Communication, **19** (8), 685-699 (2004)
25. P. van Beek, J. R. Smith, T. Ebrahimi, T. Suzuki, and J. Askelof: Metadata-driven multimedia access, Signal Processing Magazine, IEEE, **20** (2), 40--52 (2003)
26. K. Shen and E. J. Delp: A fast algorithm for video parsing using MPEG compressed sequences. In: Image Processing, 1995. Proceedings., International Conference on, pp. 252-255 (1995)
27. H. L. Wang, A. Divakaran, A. Vetro, S. F. Chang, and H. F. Sun: Survey of compressed-domain features used in audio-visual indexing and analysis, Journal of Visual Communication and Image Representation, **14** (2), 150-183 (2003)
28. J. Bescos: Real-time shot change detection over online MPEG-2 video, Circuits and Systems for Video Technology, IEEE Transactions on, **14** (4), 475-484 (2004)

29. S. Jeannin and A. Divakaran: MPEG-7 visual motion descriptors, IEEE Transactions on Circuits and Systems for Video Technology, **11** (6), 720-724 (2001)
30. Y.-P. Tan, D. D. Saur, S. R. Kulkami, and P. J. Ramadge: Rapid estimation of camera motion from compressed video with application to video annotation, Circuits and Systems for Video Technology, IEEE Transactions on, **10** (1), 133--146 (2000)
31. 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, **14** (4), 462--474 (2004)
32. D. Mukherjee, E. Delfosse, J. G. Kim, and Y. Wang: Optimal adaptation decision-taking for terminal and network quality-of-service, IEEE Transactions on Multimedia, **7** (3), 454-462 (2005)
33. T. Zgaljic, N. Sprljan, and E. Izquierdo: Bitstream syntax description based adaptation of scalable video. In: Integration of Knowledge, Semantics and Digital Media Technology, 2005. EWIMT 2005. The 2nd European Workshop on the (Ref. No. 2005/11099), pp. 173-178 (2005)
34. M. H. Chan, Y. B. Yu, and A. G. Constantinides: Variable size block matching motion compensation with applications to video coding. In: Communications, Speech and Vision, IEE Proceedings I, pp. 205-212 (1990)
35. L. Herranz, F. Tiburzi, and J. Bescós: Extraction of Motion Activity from Scalable-Coded Video Sequences, In: Semantic Multimedia, Lecture Notes in Computer Science (4306), Springer-Verlag Berlin, pp. 148-158 (2006)
36. M. Hamidi and J. Pearl: Comparison of the cosine and Fourier transforms of Markov-1 signals, IEEE Transactions on Signal Processing on Acoustics, Speech and Signal Processing, **24** (5), 428-429 (1976)
37. I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang: Video transcoding: an overview of various techniques and research issues, IEEE Transactions on Multimedia, **7** (5), 793-804 (2005)