DOI: https://doi.org/10.1016/j.image.2007.05.009

Content-driven Adaptation of On-line Video [*]

Jesús Bescós, José M. Martínez, Luis Herranz and Fabricio Tiburzi

Abstract

This work presents an on-line approach to the selection of a variable number of frames from a compressed video sequence, attending only to rules applied over domain-independent semantic features. The localization of these semantic features helps infer the heterogeneous distribution of semantically relevant information, which allows to reduce the amount of adapted data while preserving the meaningful information. The extraction of the required features is performed on-line, as demanded by many leading applications. This is achieved via techniques operating on the compressed domain, which have been adapted to operate on-line, following a functional analysis model that works transparently over both DCT-based and wavelet-based scalable video. The main innovations presented here are the adaptation of feature extraction techniques to operate on-line, the functional model to achieve independence of the coding scheme, and the subjective evaluation of on-line frame selection validating our results.

1 INTRODUCTION

One of the main addressed topics in the area of video content processing and management is the adaptation of audiovisual media to different environments. Early works approached adaptation as a content-agnostic problem, trying to match the constraints imposed by the terminal, network and user

[*] Work partially supported by the European Commission under its 6th Framework Programme (FP6-001765 - aceMedia Project) and by Spanish Institutions under projects TIN2004-07860-C02-01 and S-0505-TIC-0223.

in each session, and producing an adapted media just from a signal processing point of view (usually using the rate-distortion criteria as a quality measure). Currently, a first challenge is to achieve adaptation while maintaining most of the desired information; a second challenge is to do this in an efficient way.

Regarding the first challenge, traditional content-blind adaptation is performed via transcoding, that is, decoding the original media and then re-encoding it with different compression parameters according to the target scenario. There are some more efficient solutions like transcoding without fully decoding [1] or performing direct bitstream extraction via a Bitstream Syntax Description (BSD) [2] for scalable formats. However, both solutions are similar from the content point of view.

Current video coding standards are focused on preserving perceptually relevant information, but not so much on considering semantic relevance. Some coding features, like variable GOP size or variable size block matching, certainly allow to enhance the quality of specific sequence intervals or regions. However, most current encoders do not take full advantage of these possibilities. In conclusion, most encoded video sequences keep being blind to semantic relevance.

Semantically relevant information is highly concentrated in spatial or temporal events (i.e., when something happens) and shows very low variation elsewhere (i.e., in static scenes or frame regions, or under global and slow motion): its temporal and spatial distribution along a video sequence is highly inhomogeneous. Therefore, any coding or adaptation scheme that considers the distribution of semantic features can drastically reduce the final bitstream size without a significant content loss.

In fact, this semantic information is usually highly dependent on the specific application domain or context, ranging from the presence or absence of objects (e.g., people, faces, race cars) to more complex relationships between objects or actions (e.g., objects disappearing or unattended, people running, cars overtaking or crashing), which makes difficult to follow a generic or domain-independent approach. Nevertheless, some meaningful events can be considered independent of any specific domain (e.g., shot changes, camera motion scheme variations, object movements relative to the camera motion, etc.), and are therefore specially fit for domain-independent content-aware media adaptation. One of the objectives of the work presented here is to extract some of these semantically relevant features, specifically those related to overall frames.

Regarding the second challenge, if the aim is to adapt stored and on-line video content, the identification of relevant features should ideally be performed on-line for every adaptation request. This would avoid the requirement to store semantic hints which might be quite bulky (e.g., segmentation masks for each frame), and the need to perform an exhaustive feature extraction, which is not only costly but even useless for many adaptation operations. Additionally, a broad and increasing range of applications based on the use of live visual information (TV, surveillance, mobile, etc.) do require on-line video processing, which prevents from applying most of the existing approaches to video skimming or summarization.

The state of the art extraction of semantic features from video sequences is, in general, far from performing in real time. Moreover, techniques are not usually intended to operate on-line, that is, with a reasonable delay. This paper shows that, up to some extent, it is possible to perform content-based adaptation in an efficient way and with negligible delay. First, in order to achieve efficient operation, our feature extraction is performed with techniques that work on the compressed domain. This has clear advantages, as the direct availability of estimators for features that are hard to extract at pixel level (e.g., the motion field), and a dramatic reduction of the dimensionality of data (e.g., by working over DC images); the main drawback is that these techniques are highly dependent on the coding standard and compression parameters. Here we present a functional model aimed at obtaining features mostly independent of the specific coding scheme. Second, in order to fulfil the on-line operation requirement we present modifications of existing techniques, for instance alternatives to frame selection algorithms currently based on optimization strategies that require full sequence segments in advance.

This paper presents an on-line approach to the content-driven selection of a variable number of frames from a compressed video sequence, in order to later generate adapted media in any of the possible application modalities (e.g., adapt the original sequence to a slide show or to a skimmed video). Frame selection is organized in levels (the higher, the lesser number of frames) following a hierarchical scheme which is event-based in its higher levels and (frame-)rate-based in its lower levels. In order to validate our approach, we state the problem and show some initial and innovative experiences to the evaluation of on-line frame selection via user tests. In this sense, to the best of our knowledge, there are no works confronting the generation of a frame selection ground truth for on-line operation, that is, requesting assessors to select frames *as they are first inspecting* a video sequence.

After Section 2 on related work, Section 3 states the overall context of the work presented here and introduces the functional model that allows to transparently operate over two notably different coding schemes: DCT-based video and wavelet-based scalable video. Section 4 deepens into descriptions of the feature extraction algorithms for the two considered coding schemes. Section 5 shows the hierarchical frame selection procedure and the rules guiding it. Finally, Section 6 focuses on user evaluation and includes some experimental results of the application of this approach to different content-driven adaptation types.

2 RELATED WORK

The focus of this paper is content-driven adaptation, which is a highly interdisciplinary area. It covers topics like media adaptation architectures, feature extraction frameworks, video analysis over compressed coding domains, etc. Related work on these somehow independent topics will be presented in each corresponding Subsection. This Section addresses the overall subject of this paper, i.e., our approach to content-driven adaptation based on frame-related features (as opposed to object-related ones) and, more precisely, on hierarchical frame selection techniques from video sequences. This hierarchical approach resembles a video summary in its higher levels and a video skim in its lower ones.

A video summary (also referred to as *still abstract* or *static story-board*) consists of a set of salient images or key-frames that represent accurately the content of the original video. A video skim (also known as *moving abstract*) consists of a collection of image sequences extracted from the original video, but with a considerable shorter global length. This is achieved either by just speeding up the playback[3] or by cutting parts of the video by highlight extraction[4][5][6].

Early works in video summarization[7] selected key-frames by randomly or uniformly sampling the original sequence. This content-blind operation resulted in not representing properly short important segments and/or overrepresenting long unimportant ones. Current status of computer vision and image understanding techniques is far from building a system capable of ranking automatically the semantic relevance of video frames by interpreting and evaluating the visual content. However, it has been noticed that semantically important information is highly related with some “syntactic clues” (e.g., the camera motion) that can be obtained without an in depth understanding of the content. These clues provide us with a witty way to adapt content via key-frame selection.

Of all the many recent works on video summarization, except for some reports that deepen into post-processing of shot key-frames to remove possible inter-shot redundancy[8][9], the revised works operate within shots, assuming at least one key-frame per shot to avoid losing a priori unpredictable frames. A main observation is that most of these approaches cannot be adapted to on-line operation, as they require to take into account the whole sequence segment in advance.

Works based on parameter optimization define a quality measure over a set of key-frames and aim at finding the set that maximizes such measure for a segment. Typical quality criteria are the minimization of the cross-correlation among certain features of the key-frames (e.g., colour and motion of each frame[10], cumulated action[11]) gathered in a feature vector, or the optimization of the rate-distortion ratio[12].

Clustering based approaches[8][13][14] aim at grouping similar frames to avoid the redundancy in summaries (i.e., the presence of various frames with similar content). The required frame similarity measure is gathered in a feature vector including colour (the most popular), texture, shape of the objects, camera/object motion, or a combination of them.

As motion in a video sequence implies content evolution and change, motion features can provide useful clues for key-frame selection. In this direction, [15] uses the frame optical flow, [16] uses the MPEG-7 motion activity descriptor, and [17] computes a metric based on the macroblock prediction types from the MPEG-1/2 coding standards.

Objects are also closely related to the human perception of visual content. Exploiting this fact, [3] selects frames according either to significant variation in the number of objects in a scene, or to action change (measured as global frame variation) in the absence of the former. In [18], key-frames are selected as a function of the *objects separability*, which minimizes the errors of a later intra-frame spatial segmentation.

Visual attention approaches are being progressively considered for several video analysis tasks as an alternative way of video content understanding from the perspective of the human perception mechanism. In this direction, in [19] a user attention model is used to create an attention curve which is used as an importance ranking of the video content.

3 FRAMEWORK OVERVIEW

The frame selection approach that we show here is part of one of the available content adaptation tools of the Content Adaptation Engine presented in [20], more specifically the tool devoted to content-aware video adaptation. All the adaptation issues related to the media input-output and to the media and context descriptions are managed by the core of this Content Adaptation Engine. For completeness, this Section briefly describes the architecture and global operation of the content-aware video adaptation tool and introduces the functional model followed by one of its main modules, the one devoted to content analysis.

3.1 Content-aware video adaptation tool

This Section outlines the architecture of the adaptation tool (see Fig. 1) and some design considerations proposed to perform content-driven adaptation of video material. The main differences between this approach and other reported architectures[21][22] for content-aware video adaptation are that we focus on on-line extraction of the semantic features controlling the process, and that the control loop described here to manage the output bit-rate is mainly based on the variation of the number of included relevant frames. We rely on the use of frame-level features (e.g., shot boundaries, changes in the camera motion scheme, motion activity, etc.), although this framework also considers the use of object-level features related to inter-objects spatial and temporal relationships.

The inputs to this tool are the video source and the adaptation parameters. Video sources currently include MPEG-1/2 video and wavelet-based scalable video. Scalable Video Coding (SVC) aims at coding a video sequence so that a single encoded bitstream can be efficiently decoded at different fidelity levels. This reduces content-blind adaptation to almost a selection of the necessary parts of the bitstream. Here we will refer to a specific implementation of a Wavelet-based approach to SVC (WSVC), outlined in Section 4.1.

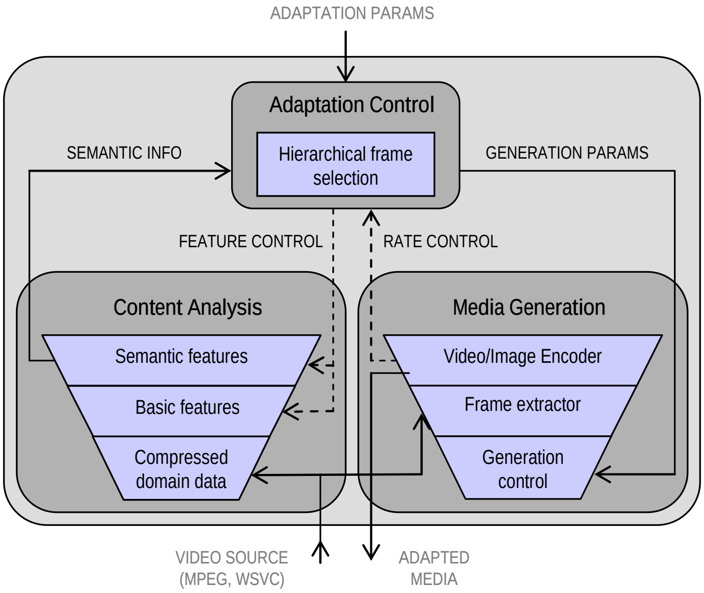


Fig. 1: Architectural diagram of the content-aware video adaptation tool.

The adaptation parameters refer to the content, described via media descriptions thanks to the MPEG-7 Multimedia Description Schemes specification[23], and to constraints externally imposed by the context or usage environment: user preferences, network characteristics and terminal capabilities. These constraints are described via profiles using the MPEG-21 Digital Item Adaptation specification[24]. Content and context descriptions are managed by the aforementioned Content Adaptation Engine, which finally calls this tool specifying a video adaptation mode (either a video skim or an image storyboard), frame size requirements, and frame and data rate limits; these are the effective adaptation parameters currently accepted by the content-aware video adaptation tool.

The three modules identified in the architecture operate as follows:

1) The Content Analysis is in charge of extracting semantically relevant features from the video source. The type and number of features extracted depends on the initial decision performed by the Adaptation Control. The extraction of these features is performed on-line, with techniques that operate on the compressed video data, following an abstraction model that eases transparent adaptation of DCT-based video and wavelet-based scalable video.
2) The Adaptation Control orchestrates the overall adaptation process. First, according to the adaptation parameters, it commands the Content Analysis to read the video source and to extract the required features. Then, starting from the extracted features and from a preliminary estimation of the target bitrate, it performs the selection of the frames that will initially shape the output media. Finally, it launches the Media Generation, establishing a loop that updates the amount of selected frames and the number of extracted features as a response to the actual output bitrate.
3) The Media Generation receives a list of frames to start generating the output media in the selected video adaptation mode. In parallel to the generation process, it measures the output bitrate in order for the Adaptation Control to update its operation.

3.2 Abstractions for visual feature extraction

Here we introduce a functional model for the analysis of video sequences. The aim is to obtain semantically relevant features which are almost independent of the specific coding scheme of the source media.

The current implementation of the Content Analysis module considers two quite different coding schemes: MPEG-1/2 video and WSVC video. In order to share the use of common analysis algorithms among several codecs, we propose the functional model described in Fig. 2. It divides the content extraction process in three stages, from the highly specific codec-dependent compressed domain data to the coding-independent semantic features. Details of the involved algorithms are further explained or referenced in Section 4.

The identified semantic frame-level features will be used as hints to guide the adaptation process. One of the hypotheses of this work is that these features can be considered to indicate, up to a certain degree of confidence, semantically relevant information which can be applied to any context or application domain. This is why we call them *semantic features*, although from a video indexing point of view they are usually called *low-level features*. Another hypothesis is that it is possible to design algorithms to extract these features in an on-line and efficient way.

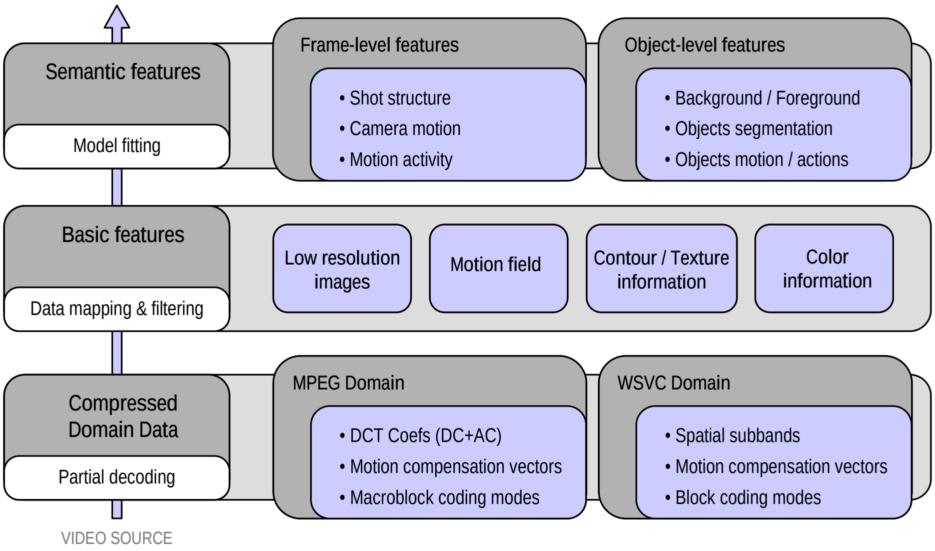


Fig. 2: Stages in the functional model for the coding independent analysis of video sources.

It is reasonable to assume a priori generic semantic relevance (i.e., domain-independent) for shot changes, as they indicate changes in the recording camera, and for changes in the camera motion scheme, as they indicate either changes in the recorded target (the camera “moves” from one target to another by panning, zooming, etc.) or absolute motion of this target (the camera moves to track the target).

Additionally, the camera motion scheme (not its changes), together with an estimation of the motion activity, may also help infer semantic changes in the image plane. Camera motion parameters provide the means to obtain an estimation of the time interval required for a frame to show a scene view with a high percentage of content difference respect to that shown by a previous frame. This fact, which has been systematically applied in the generation of panoramic views from a video sequence and also applied to key-frame selection[25], can be assumed for scene objects that do not show an average motion equal to the camera motion (i.e., usually the background and scene objects not being tracked by the camera). In order to infer content change in the remaining objects (i.e., the targets that are being tracked) or in the absence of camera motion, we rely on measures of motion activity which have proven to be a simple and efficient way to obtain indicators of the perceived scene activity[26].

For completeness, we show an analysis framework devoted to parse the frame-level and object-level structure of the video sequence. Object-level features are usually closely related to the inference of higher level semantics (e.g., actions or complex events) which usually depend on the application domain. For this reason, the work on content-aware adaptation presented in this paper deals only with the frame-level ones, that can be considered domain-independent up to some extent.

4 COMPRESSED DOMAIN FEATURE EXTRACTION

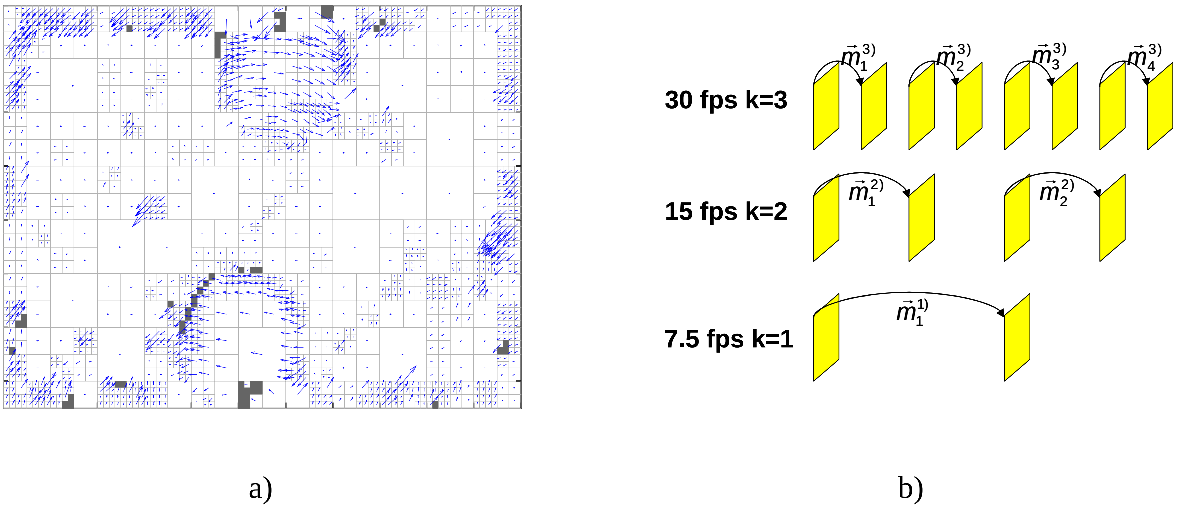
This section summarizes the algorithms included by every extraction stage of the functional model. The two first stages describe independent techniques for MPEG-1/2 video and WSVC video. Algorithms in these stages are quite established for MPEG-1/2; we just detail innovative but quite trivial approaches to obtain equivalent efficient features from WSVC. The third stage assumes the availability of basic features, independent of the coding scheme, to confront the more challenging inference of semantically relevant features, specially if on-line operation is required. Here we briefly describe and refer to previous work in this topic.

4.1 Video sources and coding schemes

We currently consider two quite different types of coded video sources. On one side, the widely known MPEG-1/2 video; on the other, wavelet coding, which enables a natural multiresolution framework for highly scalable video coding[27]. Most works on this last subject are based on two steps: first, a spatial 2D Discrete Wavelet Transform (2D DWT); second, another wavelet transform in the temporal axis, combined with motion compensation, known as Motion Compensated Temporal Filtering (MCTF). Alternatively, there are many codecs that use the so known t+2D framework or Spatial Domain MCTF (SD-MCTF), where the temporal transformation is performed before the spatial one[28][29][30]. Our functional model for analysis currently includes the scalable coding approach based on the t+2D framework described in [30].

This scalable coder operates in three stages. The first performs a MCTF which yields a hierarchical temporal decomposition of frames belonging to a Group of Pictures (GoP), similarly to the MPEG concept. A set of motion compensation vectors results from each decomposition subband, providing temporal scalability. The second consists of a 2D DWT aimed at decorrelating each temporal subband while achieving spatial scalability. The last stage performs texture coding of the resulting spatio-temporal subbands, here enabling quality scalability, and entropy coding of the scalable motion vectors. As a result, this coding scheme provides spatial, temporal and quality SNR scalability.

As opposed to most established video standards, which use a fixed-size block-matching algorithm to perform motion estimation for compensation, WSVC goes for a Hierarchical Variable-Size Block Matching scheme (HVSBM[31]), also used in the recent MPEG-4 AVC/H.264 standard[32] and in most of the MCTF frameworks for scalable video coding. The motion field is here partitioned in a quad-tree structure, and a motion vector is assigned to each block in each leaf block (see Fig. 3a).



a)

b)

Fig. 3: a) Example of the motion field that results from applying HVSBM within the WSVC coder: quad-tree structure and vectors indicating the magnitude and direction of the estimated block motion. b) Example of motion vectors in a GoP with eight frames and three decomposition levels; $\vec{m}_i^{k)}$ corresponds to the *i*-th vector of the *k*-th level.

Motion vectors in WSVC are further organized to enable temporal scalability: depending on the temporal level, the temporal distance (measured in frames of the original sequence) between the current frame and the reference frame is different. Each set of motion vectors corresponding to each temporal level represents the same motion information of the GoP, but at different scales. This is important in terms of efficiency, as the number of motion vectors to process per GoP depends on the desired level, being less in lower levels at the expense of reducing precision in the temporal axis (see example in Fig. 3b).

4.2 Compressed domain data

Fast analysis is mainly based on the use of compressed data, easily available in the bitstream, which can be used to infer content features. Just the first lightweight stages of the decoding process, usually only entropy decoding, are necessary to extract these data.

Regarding MPEG, the compressed domain data are mainly the DCT coefficients (DC and AC) and the macroblock motion vectors. Extraction of the former for every intra-coded macroblock and of the latter for every predicted macroblock only requires header parsing and VLC decoding of the video stream. The macroblock coding modes have also shown to be useful for some analysis techniques[33]. It should be noted that, although none of the presented analysis algorithms requires full decoding of any video frame, the latter media generation process will in general require full decoding of the selected frames. For this reason, depending on the requirements in terms of temporal resolution, if the algorithms can avoid selecting B and even P frames, which are the most costly to decode, a better performance will be possible. In this direction, the Adaptation Control is able to command a feature extraction based on the rate control feedback (see Section 3.1).

Regarding WSVC, we can take advantage of its scalable structure to extract only useful information, avoiding unnecessary inverse MCTF and spatial subband reconstruction stages, and just focusing on entropy decoding of the subbands. Full motion information, low resolution versions of some frames and block coding modes are also directly available in this way.

4.3 Basic features

The purpose of this stage is to get a representation of the video data which is independent of its original coding scheme. Three basic features are currently considered: low resolution images, motion field and contour/texture information. Colour information, although separately depicted in Fig. 2 for generality, is currently just considered for low resolution images.

In the MPEG case, DCT coefficients can be exactly obtained for any intra-coded macroblock. In any other case, obtaining the exact value of a coefficient requires in general four block IDCTs and one block DCT; however, this can be avoided if we assume to work with an estimation of the coefficients which can be efficiently obtained[34] at the expense of some spatial and temporal filtering which increases with the number of predicted macroblocks involved. As a result, low resolution images (i.e., DC images which are 8x8 times smaller) can be estimated directly from I frames, quite precisely for P frames, and reasonably for B frames.

AC coefficients have also proven to be useful to estimate textures and edges[35], or text captions[36]. In this sense, we use selected sets of these coefficients in two ways: first, to detect blocks containing a single edge in any direction; second, to characterize each block texture in order to detect texture change or caption blocks. We currently apply both indicators to reinforce motion-based objects segmentation. As this is not a frame-level feature, we do not here deepen into these algorithms.

The two previous features are almost coder independent. On the contrary, the motion field can be estimated from the motion estimation vectors, which are highly coder dependent. As a conclusion of our thorough work on this subject[37], we decided to balance between efficiency and vector reliability. We estimate the motion field just from the motion vectors of forward macroblocks in P frames, which yields a sparse and homogenously distributed motion field with macroblock resolution.

From a temporal point of view, this approach generates the motion field according to P frames distribution into the GoP. Further processing either to achieve higher temporal resolution or to remove outliers might be application dependent; hence, this is not considered at this stage.

In WSVC video, low resolution images can be obtained by just selecting the lower spatial resolution version of the video stream (we consider 3 spatial decompositions at encoding, so that the low resolution version is 8x8 times smaller than the original, as for MPEG DC images). These images can be directly obtained for the lower temporal resolution; higher temporal resolutions, which might be needed for some specific applications not considered at this stage, would require inverse MCTF operations.

With respect to the motion field, the WSVC implementation uses variable block size motion compensation, as described in Section 4.1. In order to obtain a common abstraction of the motion field, and considering that the MPEG one is referred to a constant block size (16x16 pixels), we include a normalization step: vector replication when the block size is bigger than the MPEG one, and vector averaging when it is smaller. Regarding temporal properties, we can directly obtain the motion field at different resolutions (the lower the more efficient), and just for odd to even frames, as depicted in Fig. 3b.

4.4 Semantic features

This stage refers to features closer to a general understanding of what is happening in the video sequence (e.g., the recording camera has changed, the camera starts zooming, objects move quickly with respect to the camera) than to signal related characteristics. As mentioned above, we will focus here on frame-related features that can be considered to be independent of the application domain.

4.4.1 Shot structure

Video shots are often used as the basic unit of video temporal segmentation, as they group frames with similar content. For video shot detection we use the algorithm proposed in [38] that works directly with frame histogram metrics computed over low resolution colour images obtained from the previous stage. This algorithm detects shot boundaries with frame precision, and achieved 0,99 recall and 0,95 precision for cut detection over a representative sample of over 2000 shot transitions. We have evaluated the behaviour of this algorithm for different spatial resolutions and quality levels of WSVC video[39]. The results show that using the version with the lowest spatial resolution, but with the original quality and frame rate, the performance in cut detection is similar to that of the MPEG case, while the processing time is much lower.

4.4.2 Camera motion

Here we use a novel two-phase technique[40] which is motivated by the efficiency requirements of on-line operation. The first phase aims at detecting changes in the camera motion scheme represented by the motion field. The identified motion changes divide shots into smaller sequence segments. The second phase aims at obtaining the motion parameters for each of segment. This is performed via fitting to an affine parameter model.

Following efficiency considerations, we suggest that the traditional frame by frame fitting can be unnecessary or even disadvantageous: first, because in many cases the dominant motion is quite regular and therefore the information obtained is highly redundant; second, because some frames are not convenient for camera motion estimation as the motion scheme in them is highly irregular. In this sense, the technique performs fitting over sets of vectors collected just from the first few consecutive frames of the segment (e.g., in the MPEG case, all the consecutive P frames of the first GoP).

The performance of this algorithm is similar to that of other state-of-the-art approaches. However, the overall efficiency is notably greater, as model fitting operations are dramatically reduced, and the two-phase approach results particularly suitable for the frame selection scheme described later.

4.4.3 Motion activity

The perceived activity of a sequence segment is a visual feature commonly used in tasks as video analysis, content retrieval or video summarization[16]. MPEG-7 defines a MotionActivity descriptor[41] aimed at capturing “the pace of the motion in the sequence, as perceived by the viewer”. The Intensity of Activity element of this descriptor is defined as the standard deviation of the magnitude of MPEG motion vectors, normalized and quantized into five levels. [26] describes a thorough work on low-complexity measures of this activity concept, computed from MPEG block motion vectors. According to the results reported there and considering that we aim to characterize the motion activity that is independent of the camera motion, we use an evolution of the *mean0* measure described in that report. Similarly to the MPEG-7 descriptor, the *mean0* measure is based on the observation that “the perceived motion activity is higher when the motion is not uniform”. Its value for a frame *n* is obtained as:

$$I_{mean0}(n) = \frac{1}{N}\sum_{i=0}^{N-1} \left\|\vec{m}_i - \bar{m}\right\|^2 , \quad \bar{m} = \frac{1}{N}\sum_{i=0}^{N-1} \vec{m}_i , \qquad (1)$$

where N is the number of motion vectors, $\vec{m}_i$ is the *i*-th motion vector and $\bar{m}$ is the average.

As we have previously estimated the camera motion, it is possible to use this value instead of the average of motion vectors. This motion activity calculation over camera-motion compensated vectors is ideally equal to the *mean0* measure for translational camera motions, but would much more precisely model the above observation for camera rolls and zooms (as for these motion patterns the average of motion vectors would ideally be zero in the absence of moving objects with respect to the camera). We will refer to this new motion activity feature as:

$$I(n) = \frac{1}{N}\sum_{i=0}^{N-1} \left\| \vec{m}_i - \vec{c}_i \right\|^2 , \tag{2}$$

where $\vec{c}_i$ is the component of the $\vec{m}_i$ vector due to the camera motion, which has been previously obtained in the second phase of the algorithm presented in Section 4.4.2.

This motion activity feature is calculated over the motion field available as a basic feature for every targeted coding scheme (see Section 4.3). In the MPEG case, N is the number of frame macroblocks and Eqn. (2) is computed for every P frame. In the WSVC case, we estimate a similarly distributed motion field (with the MPEG macroblock resolution, and hence the same value of N), but we can obtain it for every temporal level (see Fig. 3b). Hence, we can compute this motion activity feature for any of these levels.

5 HIERARCHICAL FRAME SELECTION

Frame selection is guided by the localization of frame-level relevant events that can be considered domain-independent. However, it is evident that the amount of selected frames should vary depending on the application. In the case of an image storyboard, either the end user or the terminal decides on the maximum number of accepted frames, or on the frames per temporal unit. The frames to be included in (or removed from) a video skim depend on the available bit-rate, and hence should vary according to this parameter. In order to cope with all these situations, the selection process requires to establish some kind of scheme allowing to select as many relevant frames as required, and some sort of priority policy to guide which specific relevant frames to select.

According to the domain-independent semantic information available, we have identified up to four priority levels (see Fig. 4): the higher the level, the greater the number of selected frames. The first three levels base frame selection on the presence or absence of semantically relevant events; hence, the number of selected frames into these levels is fixed by the number and type of the semantic events in the sequence. On the contrary, the fourth level selects a variable number of frames by adjusting the degree of magnitude sub-sampling of another semantically meaningful feature: the motion activity. As a result, we present a procedure to select an increasing number of relevant frames, which ideally results in an homogeneous subsampling of the semantically relevant information. The following subsections provide details on the selection policies applied for each level.

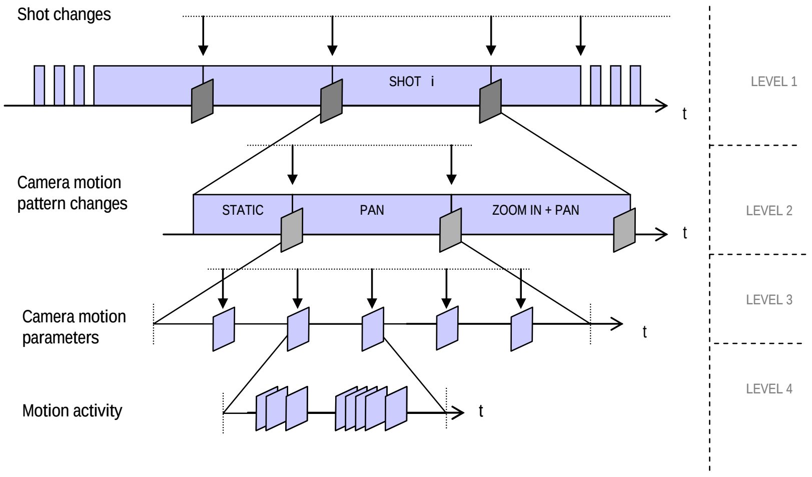


Fig. 4: Hierarchical frame selection according to the presence of semantically relevant events (levels 1 to 3) and to the progressive inclusion of semantically relevant information (level 4).

5.1 Level 1: Shot based selection

Let us consider the general case of a video sequence containing several shots. As the video source is being analyzed, the shot change detector marks shot boundaries with frame precision; these correspond to the first priority level, which is event based.

Frame selection, when only considering this level, is a topic largely reported in early works on key-frame selection, which is usually based on cut detection only. Criteria for key-frame selection ranges from the first frame after the cut to either the tenth (to avoid instabilities around the video edition effect) or to the first satisfying some given image quality condition. In the absence of an established criterion and taking into account efficiency considerations, we select the first frame after the shot boundary that can be efficiently extracted (that is, the first I frame for MPEG and the first lower temporal subband frame from WSVC).

5.2 Level 2: Selection based on changes in the camera operation scheme

Let us now consider a single shot segment as the set of frames between the first selected shot frame in Level 1 and the first frame of the following shot boundary (so that frames belonging to a gradual transition are never selected). This segment generally shows different successive camera motion schemes. The first phase of the camera motion detection algorithm presented in Section 4.4.2 will operate on-line to indicate changes in the motion pattern (with P frame precision for MPEG and with selectable precision for WSVC); these correspond to the second priority level, which is also event based.

As opposed to a shot change, in this case the image plane keeps capturing the same scene view; hence, the last frame of a segment defined by a motion pattern will be almost identical to the first frame of the following segment (unless the camera shot changes, which would launch a level 1 frame selection). Additionally, as some authors have reported[15], directors frequently use camera motions to move from one location to another to show the connection between the two events. Following these observations we select the last frame of the homogeneous motion segments (i.e., the frame at which the camera *decides* to terminate this motion and begin with another).

5.3 Level 3: Camera motion based selection

This level assumes the availability of homogeneous sequence segments from the camera motion point of view. The second phase of the camera motion detection algorithm presented in Section 4.4.2 will also operate on-line to indicate the type of motion pattern for this segment (e.g., pan, zoom) along with the six parameters that characterize the affine model used to identify it.

Since in generic video sequences certain motion restrictions can be usually assumed (far objects, small motion in the Z axis, small inter-frame angular motion, etc.), we currently consider only pan, tilt, zoom and roll camera motions. Additionally, due to the precision of current algorithms we consider just one of these motions for every segment, instead of a combination of them.

Camera motion model	$\begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_0 \\ b_0 \end{bmatrix}$	d_x, d_y are the horizontal and vertical components of the motion vectors and x, y are the image plane coordinates, all measured in pixels.
Other parameters	W, H are respectively the frame width and height. T_f is the time interval (measured in frames) corresponding to the available motion vectors. F_n is the frame novelty fraction.	
Motion scheme	Pan, Tilt, Track, Boom	Zoom
Assumptions	$a_1 = a_2 = b_1 = b_2 = 0\,,\;\; a_0, b_0 \neq 0$	$a_1 = b_2 \neq 0\,,\;\; a_2 = b_1 = a_0 = b_0 = 0$
Event time stamps (t_i measured in frames)	$t_i \in \mathbb{Z} \Big/ \; i \cdot F_n \leq \left\| \left(\frac{a_0}{W} + \frac{b_0}{H} \right) \frac{t_i}{T_f} - a_0 b_0 \left(\frac{t_i}{T_f} \right)^2 \right\|$	$t_i \in \mathbb{Z} \Big/ \; i \cdot F_n \leq \left\| 1 - (a_0 + 1)^{\frac{2t_i}{T_f}} \right\|$

Table 1: Summary of the criteria used to set the events that guide frame selection in Level 3.

As introduced in Section 3.2, the selection criterion is based here on the evaluation of the amount of (dis)appearing content in the image plane, just due to the camera motion. Once we select this quantity (indicated via a frame novelty fraction, F_n), we use the motion parameters to obtain the temporal location of the *content-change* events for the overall sequence segment. As summarised in Table 1, in case of a translational-like constant motion, this location increases linearly; during a roll or in the absence of motion, no events are set (as there is no content change); and in case of a zoom the location increases exponentially.

From a rigorously semantic point of view, the F_n value should be close to 1, i.e., no content overlap in two consecutive selected frames. However, the observation of frames selected in this way is quite annoying, as the visual connection from one frame to the following is lost. The work presented in [25] suggests the use of $F_n = 0.3$. In our approach, considering the possibility of further selecting frames in Level 4, we have set a value $F_n = 0.7$.

5.4 Level 4: Motion activity based selection

This level aims at selecting frames in sequence segments delimited by selected frames of the previous levels. The frame selection criterion for this level is based on an evolution of the approach presented in [16], which suggested the idea to select frames according to a linear subsampling of the cumulated activity curve of a considered segment, instead of to a linear temporal subsampling. A similar approach was previously reported in [11] to optimize key-frame selection based on an *action* curve.

Our approach applies this idea over a camera motion compensated activity feature, and adapts it to operate on-line, that is, not requiring the overall cumulated activity curve. First, as a level 3 event is reached, the algorithm devoted to motion activity estimation starts computing the $I(n)$ feature defined in Eqn. (2) and obtaining the cumulated activity curve, which is interpolated to show a value for all the frames of the segment. Then, as the curve reaches a selectable value or level parameter (I_0 in Fig. 5) that can be varied for each sequence segment, a frame is selected and the process starts cumulating activity again.

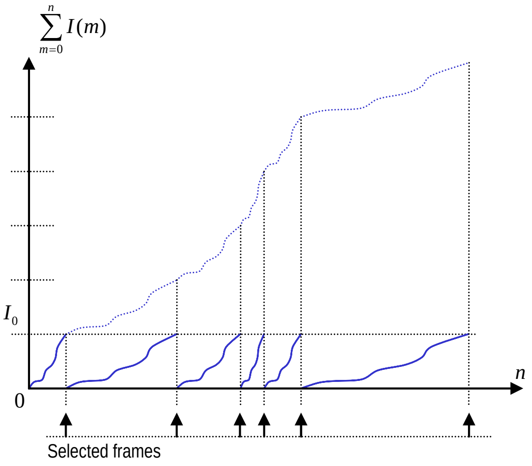


Fig. 5: Frame selection criteria based on a linear subdivision of the amount of cumulated motion activity.

According to the referenced works on motion activity measures, it is reasonable to assume that this method would yield a set of frames distributed accordingly to the perceptual motion pace of objects that move relative to the camera. This is somehow reinforced by the fact that camera motion has been independently considered in an objective way. However, a rigorous validation of this assumption should be undertaken via subjective tests, as detailed in Section 6.

Throughout this Section we have described several hierarchically organised criteria for the selection of a variable number of frames according to semantic information. Although setting the fourth level I_0 parameter to zero would result in selecting almost all the frames in the sequence, this is not the aim of this last level, as it would dismiss any semantic motivation. If we further desired to select frames due to bit-rate availability, a fifth signal based level (as opposed to a semantically based one) could be included. This could select frames from the fourth level segments either by linearly subsampling or based on some of the reported rate-distortion techniques[12].

6 EVALUATION AND RESULTS

As described in Section 2, many rigorous works on video summarization address this task as an optimization problem. This requires the definition of a quality or distortion measure, and guarantees that the achieved summarization (i.e., the resulting set of selected frames) is optimal over a sequence segment from that point of view. However, it is generally accepted that an evaluation of the degree of accuracy of a set of selected frames as representative of the content of a video segment, should be carried out via subjective tests.

The goals of multimedia content summarization are two-fold: to capture the essence of content in a succinct manner and to provide top-down access into content for browsing. The main goals are not content analysis and summarization themselves, but providing the users with informative and/or accessing improvements in their video watching experience[42]. Both aspects are covered by our approach: the different levels provide a top-down access, and each level aims to include the most different new frames, therefore increasing progressively the essence (semantic value) of the content.

Regarding essence capturing, methods for subjective evaluation of video summaries can be categorized[43] in intrinsic and extrinsic. Intrinsic methods focus on the summarization algorithm, and yield precise evaluation measures based on the analysis of the summary respect to a user-generated ground truth which, except for event-driven genres (in which summaries become highlights[44]), may be hard to obtain. Extrinsic ones focus on the user expectative, that is, on the impact of the summary on the performance of a specific task (i.e., the user ability to answer questions about the original video content); they directly target the summarization goal, but they are application dependent. In this direction, some authors have also stressed and analyzed the influence of the end-user interface on the global summarization process[42][45].

In order to perform an application-independent evaluation of our approach, we have performed an intrinsic evaluation. There are many reports on this specific topic (a thorough revision and work can be found in [46][43]). They all request a pool of assessors to somehow evaluate the absolute or relative representativeness of a set of frames, or to previously select this set in order to obtain a ground truth. As expected, assessors perform this task *after* a close inspection of the video sequence under consideration, that is, with a previous knowledge of the overall sequence.

However, the approach that we intend to validate has been specifically motivated and designed to operate on-line, that is, in a causal way. To the best of our knowledge, there are no works having requested assessors to select frames *as they are first inspecting* a video sequence. This has been the starting point for the validation process described below.

6.1 Tests description and data collection

The objective of the user tests was to generate ground truth information for the considered situation, so that no new subjective tests were required for future enhancements or new approaches. In part as a first approach to test this innovative evaluation methodology and in part to somehow allow the reader to self evaluate the results, we have selected three short and well known sequences (not for the assessors): *Akiyo* (low activity), *Foreman* and *Stefan* (medium-high activity, with panning, tilting and zooming effects). Each sequence has 300 frames with CIF resolution at 30 frames per second.

A simple application was designed for the assessors to select frames. They were required to perform frame selection in two stages: a first stage in which they should select the frames considered necessary to capture a *general idea of the sequence* without any detail, and a second stage in which frames should be selected with *every new conceptual detail*. It is worth noting the difficulty to define intermediate stages (in fact we initially defined one that had to be discarded later because half of the assessors interpreted it one way and the other half the opposite way). No indications nor constraints on the expected number of selected frames or selection pace were given. A total of 28 subjects, most of them undergraduate students, participated in this experiment.

The sources of the evaluation application, the collected data for each user, sequence and stage, and the data obtained by the frame selection algorithm in order to compare are available at http://www-gti.ii.uam.es/publications/ContentDrivenAdaptationOfOnLineVideos/Evaluation.

It should be noted that the selected sequences do not include shot transitions, as after preliminary tests we confirmed that the selection of a frame for a shot change event was guaranteed. Therefore, the collected data was aimed at validating the remaining three levels of the hierarchical selection approach.

6.2 Ground truth processing

The aim of this phase was to synthesize a reference ground truth, for each of the two considered stages, that reflected the mainstream opinion among all assessors. For this purpose we first prefiltered collected data for each of the three sequences and each of the two stages (i.e., for every data series, from now on): we discarded the data series that showed more than one standard deviation around the mean number of selected frames.

Then, the approach was to test for every data series the existence of a set of frames that was statistically preferred by the assessors. In this direction, we first decided on the number of frames for each series, n_s, based on the median of the distribution of this variable (see Fig. 6). Then we performed hierarchical clustering over the data, and selected just the centroids belonging to the larger clusters with a maximum accepted dispersion (which we set to $\sigma = 5$ frames). Fig. 7 shows the resulting selected frames ground truth for five of the six considered data series (in the stage 1, for the *Akiyo* sequence it was not possible to obtain any valid cluster, that is, to conclude on any preference).

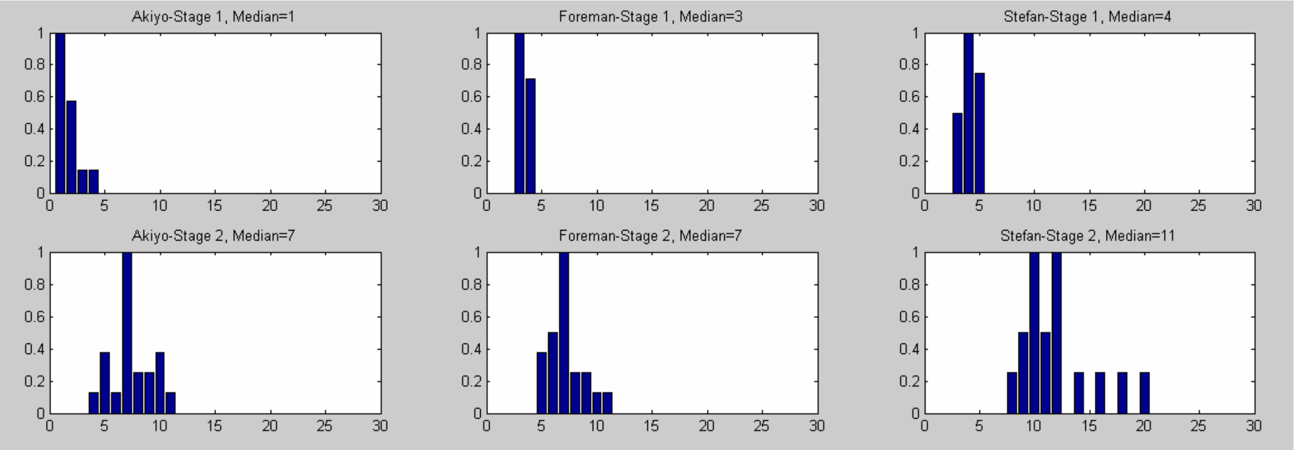


Fig. 6: Normalized histograms of the number of frames selected by each assessor for every sequence and for every stage, along with the median value of each distribution.

6.3 Algorithm evaluation

The proposed algorithm selects an increasing number of frames as more levels of the hierarchy are considered. Conversely, the ground truth extracted from the collected user data includes just two frame sets for each sequence. A thorough quantitative evaluation would require a much larger pool of assessors and, more importantly, a careful definition of more stages to guide frame selection by them. This would allow to establish correspondences between user selection stages and algorithm selection levels. However, some qualitative conclusions can be obtained, which encourage us to deepen into the presented approach and the described evaluation scheme.

In the absence of the aforementioned correspondences, we have compared each set of frames from the ground truth with the output of the algorithm that selects the same or more similar number of frames for the same sequence. This just allows to validate that the temporal distribution of the frames selected by the assessors can be obtained by some operation point of our algorithm. In this sense, Fig. 7 depicts a timeline comparison, indicating in each case the specific level of the algorithm (if level 4 was required, also an indication of the I_0 parameter is provided) designated for comparison. We can extract some observations from this Figure.

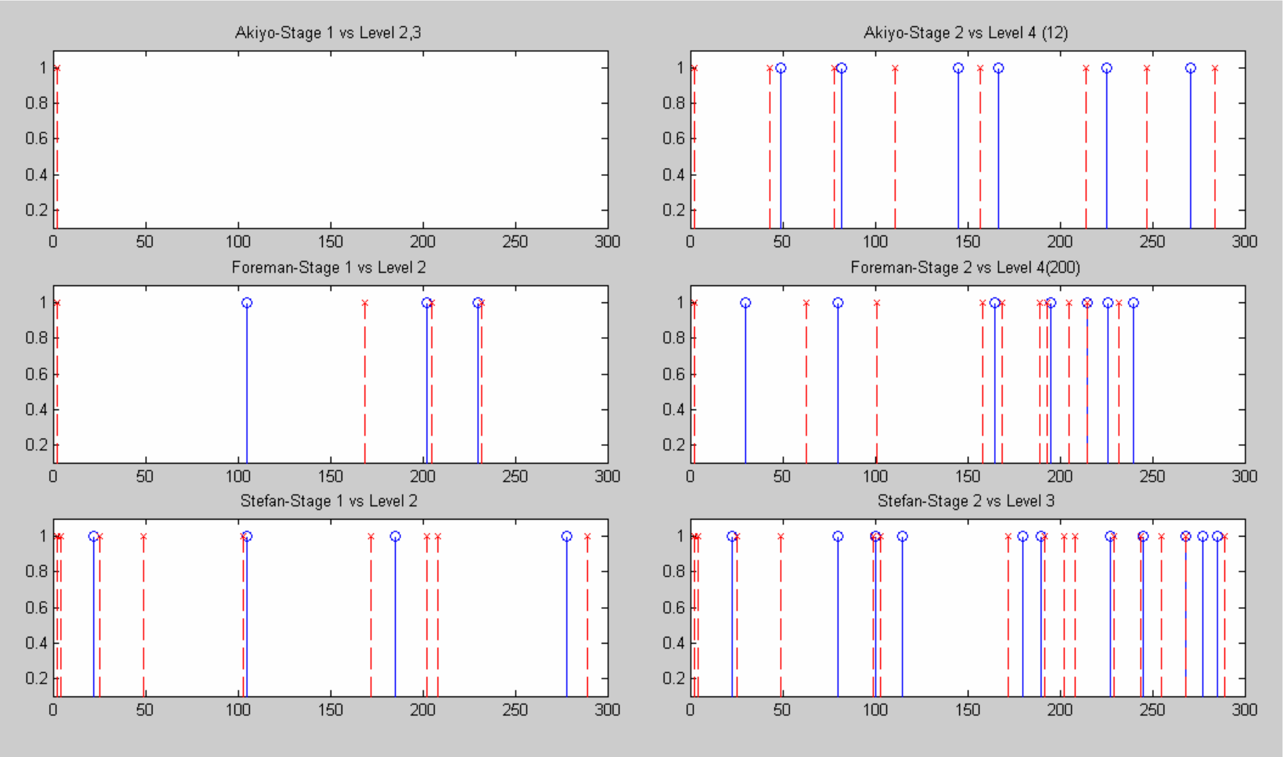


Fig. 7**: Ground truth of the frames selected by the assessors for each sequence and for each stage (circles), and frames selected by the proposed algorithm (crosses-dashed lines). The level of the hierarchy used for comparison is indicated in each graphic's title.**

Regarding level 2 operation, we can observe that well defined changes in the camera motion scheme (see the crosses in 'Foreman-Stage 1') are selected by the users (the cross around frame 170 is a false positive of our algorithm). The same applies for the most relevant changes in 'Stefan-Stage 1'; here, slight changes in the camera motion scheme have not been marked by assessors, which suggests that the threshold in the corresponding feature extraction algorithm could be increased.

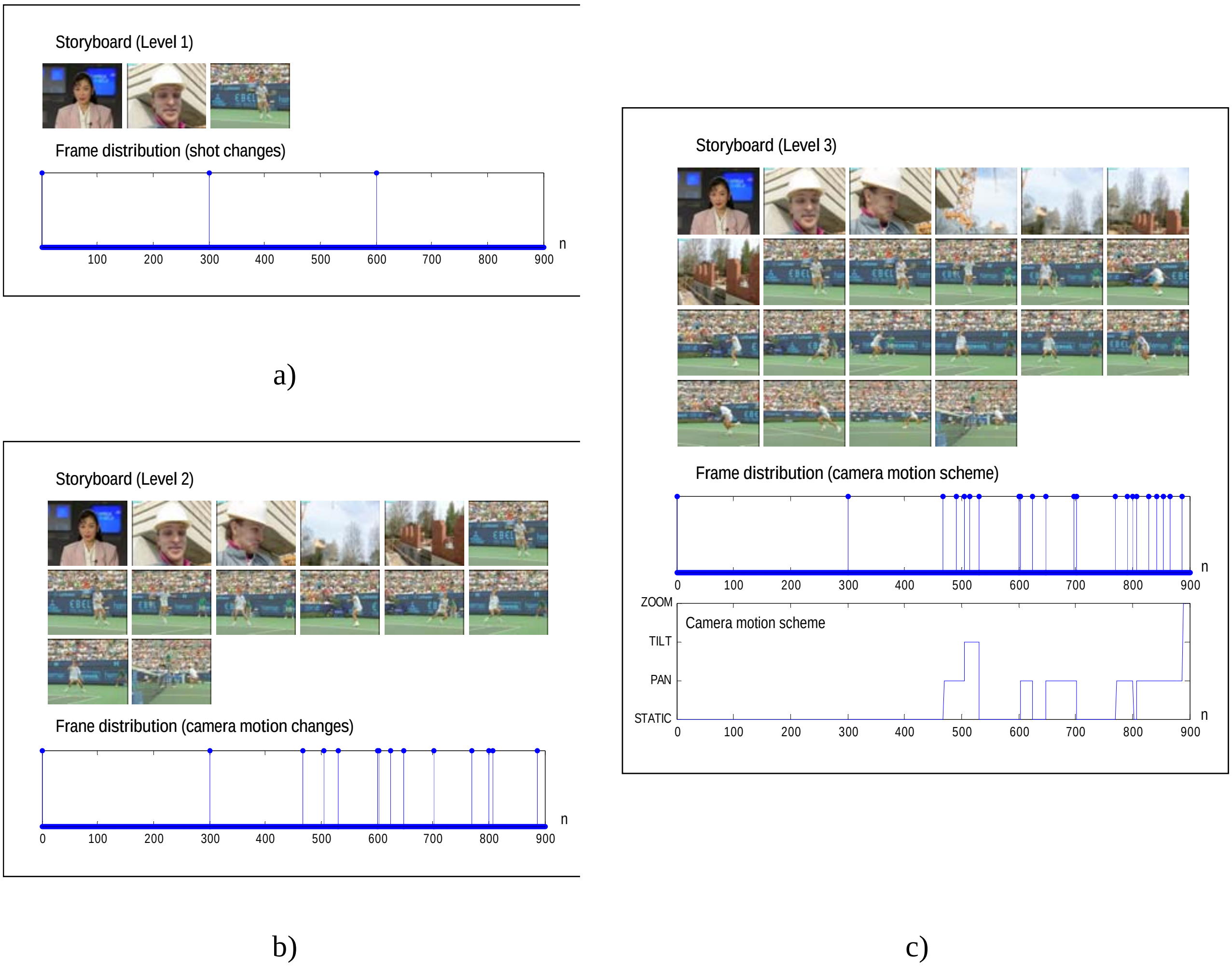
Regarding level 3 (see ‘Stefan-Stage 2’), the diagram shows a good adaptation of the frame selection algorithm to the parts of the sequence selected by the assessors (there are even several coincidences between the ground truth and the computed data).

Finally, regarding level 4, ‘Foreman-Stage 2’ and ‘Akiyo-Stage 2’ show extremes of the range of operation of the selection algorithm: $I_0 = 200$ represents the lower part of level 4 (close to level 3), and $I_0 = 12$ forces the algorithm to select frames with very slight changes in the object activity. In both cases, the matching between the computed frame locations and the ground truth is more than reasonable.

6.4 Application examples and results

This subsection shows the use of the hierarchical frame selection approach to create a video summary and a video skim from a sequence which consists of a merging of the three test sequences. It aims just at offering a friendly visual summary of some of our results.

A video summary is composed by a set of independent representative images (i.e., a storyboard). We have generated summaries for the first three levels of the hierarchy, that is, for the event based levels (see Fig. 8). Our video skim is a video with the same duration as the original, generated via on-line selection of a frame which is replicated until a new selection is performed. This results in a video with reduced content and with a reduced bit-rate. In order to offer bit-rate comparisons we have coded all the video sequences (the original one and all the skimmed sequences) with the same fixed quantification parameters and, consequently, with variable bit-rate; we then obtain a mean bit-rate.



a)

b)

c)

Fig. 8: Resulting storyboards and distribution of the selected frames for the first three levels of the hierarchy. a) Level 1: frames corresponding to the three identified shot changes. b) Levels 1 and 2: additional frames corresponding to the 11 identified changes in the camera motion scheme. c) Levels 1, 2 and 3: additional 8 frames corresponding to frame novelty due to camera motion.

Fig. 9 shows this application of the fourth level of the hierarchy. It summarizes the results (in terms of achieved bit-rate and selected frames) when varying the I_0 parameter in the range $[1, 200]$.

The quality of these videos* is increasingly acceptable as the I_0 parameter is reduced. In fact, as I_0 can be dynamically modified, it is the basis of the external rate control mechanism to on-line adapt the video skim to either quality or rate constraints (see Section 3.1).

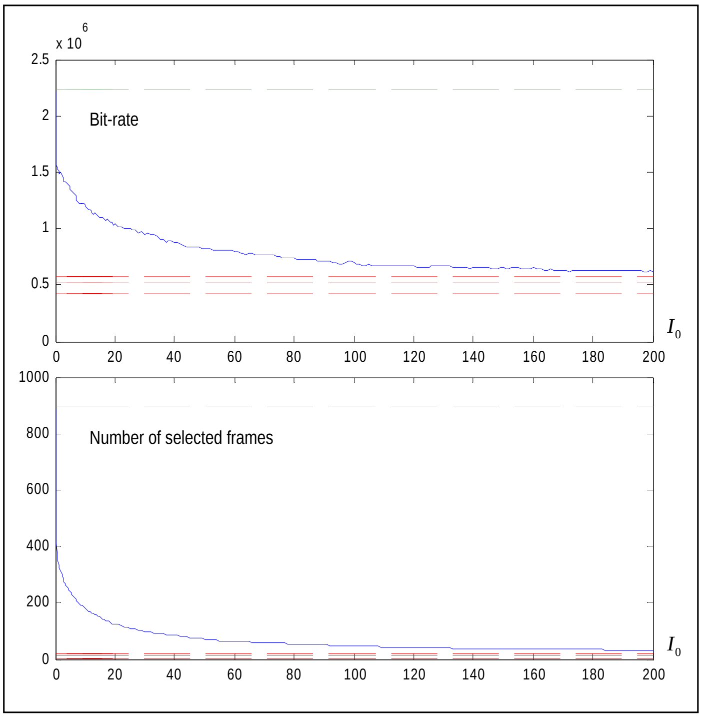


Fig. 9: Number of selected frames (bottom) and average bit-rate (top) of the video skims generated as the I_0 parameter of the 4th level is varied. The top horizontal line of each diagram corresponds to the original coded sequence and the lower ones (from top to bottom) to the third, second and first level respectively.

Regarding efficiency, on-line operation requires the set of feature extraction techniques and the hierarchical frame selection algorithm to operate in real time, which is the case. All the algorithms

* all available at http://www-gti.ii.uam.es/publications/ContentDrivenAdaptationOfOnLineVideos/

here presented have been implemented on C++ and operate faster than real time on general purpose equipment.

7 CONCLUSIONS

In this paper we have described a novel approach to the selection of a variable number of frames from a compressed video sequence, attending only to selection rules applied over domain-independent semantic features. This approach is the basis of a content adaptation tool currently operating in the Content Adaptation Engine presented in [20].

We have described a functional model aimed at unifying video analysis over compressed domains, and how we have used it on top of two different coding schemes: MPEG-1/2 video and a specific approach to wavelet-based scalable video including different approaches to block matching and motion vector schemes than those of MPEG. This illustrates the generality of the approach. According to the stages identified in the functional model, we have described a series of algorithms for on-line and efficient extraction of frame-related features which can be considered, up to some extent, to be independent of the application domain.

These features are used to develop a hierarchical frame selection scheme which considers semantic relevance in video sequences at different levels. The higher levels provide event based selection of a relatively reduced set of frames (resembling the so known *key-frames*), which makes them useful for video summary applications. The lowest level manages the selection of a variable number of frames, which makes it adequate for video skim applications.

Finally, in order to validate our approach we have undertaken an innovative evaluation approach, focused on requesting assessors to *on-line* select frames as they inspect a video sequence for the first time, and on synthesizing a ground truth. Future work is required to achieve a rigorous quantitative validation. However, the qualitative observations encourage us to deepen into the presented approach and the described evaluation scheme.

REFERENCES

[1] A. Vetro, C. Christopoulos, H. Sun, “Video Transcoding Architectures and Techniques: An Overview”, IEEE Signal Processing Magazine, vol. 20 (2):18-29, 2003.

[2] S. Devillers, C. Timmerer, J. Heuer, H. Hellwagner, “Bitstream Syntax Description-Based Adaptation in Streaming and Constrained Environments”, IEEE Transactions on Multimedia, vol. 7(3):463-470, 2005.

[3] C. Kim, J.N. Hwang, “An integrated scheme for object-based video abstraction”, Proc. 8th ACM Multimedia, pp 303-311, 2000.

[4] B. Li, I. Sezan, “Event detection and summarization in American football broadcast video”, Proc. SPIE Vol. 4676, Storage ad Retrieval for Media Databases, pp. 202-213, 2002.

[5] C. Taskiran, A. Amin, D. Ponceleon, E. Delp, “Automated video summarization using speech transcripts”. Proc. SPIE Vol. 4676, Storage ad Retrieval for Media Databases, pp. 371-382, 2002.

[6] K. Fujimura, K. Honda, K. Uehara, “Automatic video summarization by using color and utterance information”. Proc. IEEE International Conference on Multimedia and Expo, pp. 49-52, 2002.

[7] M. Mills, “A magnifier tool for video data”, Proc. ACM Human Computer Interface, pp 93-98, 1992.

[8] A. Girgensohn, J. Boreczky, “Time-Constrained Keyframe Selection Technique”, Multimedia Tools and Applications, 11:347-258, 2000.

[9] S. Lee, M.H. Hayes, “A fast clustering algorithm for video abstraction”, Proc. International Conference on Image Processing, Vol. 2:563-566, 2003.

[10] N.D. Doulamis, A.D. Doulamis, Y. Avrithis, S.D. Kollias, “A stochastic framework for optimal key frame extraction from MPEG video databases”, Proc. IEEE 3rd Workshop on Multimedia Signal Processing, pp. 141 – 146, 1999.

[11] R.L. Lagendijk, A. Hanjalic, M. Ceccarelli, M. Soletic, E. Persoon, “Visual Search in a SMASH System”, Proc. ICIP ’96, pp. 671-674, 1996.

[12] L. Zhu, G.M. Schuster, A.K Katsaggelos, B. Gandhi, “Rate-distortion optimal video summary generation”, IEEE Transactions on Image Processing, 14(10):1550-1560, 2005.

[13] A. Hanjalic, H.J. Zhang, “An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis”, IEEE Transactions on Circuits and Systems for Video, 9(8):1280-1289, 1999.

[14] Y. Zhuang, Y. Rui; T.S. Huang, S. Mehrotra, “Adaptive key frame extraction using unsupervised clustering”, Proc. International Conference on Image Processing, Vol. 1, pp. 866-870, 1998.

[15] W. Wolf, “Key frame selection by motion analysis”, Proc. IEEE Int’l Con. On Acoustics, Speech and Signal Proc., 1996.

[16] K. A. Peker, A. Divakaran and H. Sun, “Constant pace skimming and temporal sub-sampling of video using motion activity,” Proc. IEEE Int’l Conf. on Image Processing, Thessaloniki, Greece, 2001.

[17] J. Calic.; E. Izquierdo, ”Efficient key-frame extraction and video analysis”, Proc. International Conference on Information Technology: Coding and Computing, pp. 28-33, 2002.

[18] X. Song, G. Fan, “Key-frame Extraction for Object-based Video Segmentation”, Proc. International Conference on Acoustics, Speech and Signal Processing, 2005.

[19] Y.F. Ma, X.S. Hua, L. Lu, H.J. Zhang, “A generic framework of user attention model and its application in video summarization”, IEEE Transactions on Multimedia, 7(5):907-919, 2005.

[20] J.M. Martínez, V. Valdés, J. Bescós, L. Herranz, “Introducing CAIN: a Metadata-driven Content Adaptation Manager Integrating Heterogeneous Content Adaptation Tools”, Proc. 6th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2005, 2005.

[21] A. Vetro, H. Sun, Y. Wang, “Object-Based Transcoding for Adaptable Video Content Delivery”, IEEE Transactions on Circuits and Systems for Video Technology, 11(3):387-401, 2001.

[22] A. Cavallaro, O. Steiger, T. Ebrahimi, “Semantic Video Análisis for Adapted Content Delivery and Automatic Descriptions”, IEEE Transactions on Circuits and Systems for Video Technology,, 15(10):1200-1209, 2005.

[23] ISO/IEC 15938-5, Information Technology – Multimedia Content Description Interface – MPEG-7 Part 5: Multimedia Description Schemes, 2001.

[24] ISO/IEC 21000-7, Information Technology – Multimedia Framework – Part 7: Digital Item Adaptation, 2004.

[25] H.J. Zhang, J. Wu, D. Zhong, S.W. Smolliar, “ An integrated system for content-based video retrieval and browsing”, Pattern Recognition, 30(4):643-58, 1997.

[26] K.A. Peter, A. Divakaran, "Framework for Measurement of the Intensity of Motion Activity of Video Segments", Journal of Visual Communications and Image Representation, 14(4), 2003.

[27] J.R. Ohm, “Advances in Scalable Video Coding”, Proc. of the IEEE, 93(1):42-56, 2005.

[28] K. Shen, E.J. Delp, "Wavelet Based Rate Scalable Video Compression", IEEE Transactions on Circuits and Systems for Video Technology, 9(1):109-122, 1999.

[29] Peisong Chen, John W. Woods, “Bidirectional MC-EZBC with lifting implementation”, IEEE Transactions on Circuits and Systems for Video Technology, 14(10):1183-1194, 2004.

[30] N. Sprljan, M. Mrak, G.C.K. Abhayaratne, E. Izquierdo, “A Scalable Coding Framework For Efficient Video Adaptation”, Proc. 6th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS, 2005.

[31] M. Chan, Y. Yu, and A. Constantinides, “Variable size block matching motion compensation with applications to video coding”, Proc. Of IEE., 137(4): 205-212, 1990.

[32] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC: Advanced Video Coding for Generic Audiovisual Services, 2003.

[33] H. Wang, A. Divakaran, A. Vetro, S.F. Chang, H. Sun, “Survey of compressed-domain features used in audio- visual indexing and analysis”, Journal of Visual Communication and Image Representation, 14:150-183, 2003.

[34] B. Yeo, B. Liu, “Rapid Scene Analysis on Compressed Videos”, IEEE Transactions on Circuits and Systems for Video Technology, 5(6):533-544, 1995.

[35] M.L. Jamrozik, M.H. Hayes, “A Compressed Domain Video Object Segmentation System”, Proc. International Conference on Image Processing, ICIP 2002, 2002.

[36] Y. Zhong, H. Zhang, A.K. Jain, “Automatic Caption Localization in Compressed Video”, IEEE Transactions on Pattern Analisys and Machine Intelligence, Vol. 22, No 4, 2000.

[37] L. Herranz, J. Bescós, “Reliability based Optical Flow Estimation from MPEG Compressed Data”, Proc. International Workshop on Very Low Bit-rate Video, Sardinia, 2005.

[38] J. Bescós, “Real-time Shot Change Detection over On-line MPEG-2 Video”, IEEE Transactions on Circuits and Systems for Video Technology, 14 (4): 475-484, 2004

[39] L. Herranz, J. Bescós, “On the Effect of Resolution and Quality Scalability over Shot Change Detection”, under revision; available on demand.

[40] F. Tiburci, J. Bescós, “Camera Motion Analysis in on-line MPEG Sequences”, accepted for presentation in WIAMIS’07, 2007.

[41] ISO/IEC 15938-3: Information Technology – Multimedia content description interface – MPEG-7 Part 3: Visual, 2002.

[42] C. Forlines, K.A. Peker, A. Divakaran, “Subjective Assesment of Consumer Video Summarization”, Proc. SPIE Vol. 6073, Multimedia Content Analysis, Management and Retrieval, pp. 170-177, 2006.

[43] C.M. Taskiran, “Evaluation of Automatic Video Summarization Systems”, Proc. SPIE Vol. 6073, Multimedia Content Analysis, Management and Retrieval, pp. 178-187, 2006.

[44] L. Xing, Q. Huang, Q. Ye, A. Divakaran, “Subjective Evaluation Criterion for selecting affective features and modeling highlights”, Proc. SPIE Vol. 6073, Multimedia Content Analysis, Management and Retrieval, pp. 188-195, 2006.

[45] M.G. Christel, “Evaluation and User Studies with Respect to Video Summarization and Browsing”, Proc. Of SPIE Vol. 6073, Multimedia Content Analysis, Management and Retrieval, pp. 196-210, 2006.

[46] M. Huang, A.B. Mahajan, D.F. DeMenthon, “Automatic Performance Evaluation for Video Summarization”, LAMP Laboratory – University of Maryland, Technical Report-TR-144, 2004 (available at:http://lampsrv01.umiacs.umd.edu/pubs/TechReports/LAMP_114/LAMP_114.pdf)