# A framework for scalable summarization of video

Luis Herranz, José M. Martínez

***Abstract*—Video summaries provide compact representations of video sequences, with the length of the summary playing an important role, trading off the amount of information conveyed and how fast it can be visualized. This paper proposes scalable summarization as a method to easily adapt the summary to a suitable length, according to the requirements in each case, along with a suitable framework. The analysis algorithm uses a novel iterative ranking procedure in which each summary is the result of the extension of the previous one, balancing information coverage and visual pleasantness. The result of the algorithm is a ranked list, a scalable representation of the sequence useful for summarization. The summary is then efficiently generated from the bitstream of the sequence using bitstream extraction.**

***Index Terms*—summarization, scalability, scalable summary, hierarchical clustering, ranking, compressed domain**

## I. Introduction

VIDEO SUMMARIZATION techniques try to provide the user with a compact visual representation (summary) containing enough information to obtain a quick idea of what happens in the video, making easier time-consuming tasks such as video search and browsing. Most video summarization techniques build summaries by selecting some frames of the input sequence, and then present them in a specific format, leading to different modalities, such as storyboards or video skims.

Most research in video summarization has been focused on the analysis of the video sequence, with techniques trying to get some insight about the content of the video sequence. The level of understanding ranges from low level features, such as color or texture, to high level semantic concepts. The summary is then created by removing the semantic redundancy. [1] provides a comprehensive classification and review of most summarization techniques. However, current research in video summarization also addresses new functionalities and applications. Some examples are customized summaries[2], online summarization[3] or hierarchical summaries[4].

In this work we deal with summaries with another specific functionality: they are scalable. Scalability has been used in many contexts and particularly in video coding. In that context, it allows to remove parts of the bitstream while the remaining bitstream is still valid, containing a completely decodable version with lower resolution, quality or frame rate. In scalable coding, encoding is performed once, while many versions can be extracted from the bitstream, according to the specific needs of each case.

However, the concept of scalability can be also used in video summarization in a completely different sense, as a new property of the summaries themselves[4]. In this case, the scale is related to the length of the summary (e.g. duration, number of images). Depending on the case, a summary of a suitable length can be obtained without any further analysis. As in video coding, we can find many applications in video retrieval, adaptation and personalization. For instance, depending on the display size, the length of a storyboard summary can be easily adjusted to fit the available size.

In this article we address the concept of scalability in the context of video summarization, and the influence of the length over the summarization approach. In contrast to most techniques, designed to create a single summary on a short range of summarization ratios, a novel growing technique is proposed to generate, in a single pass, a scalable set of summaries, using an iterative ranking procedure, in a graceful manner and with fine granularity. Besides, an adequate framework and scalable representation (ranked list) are also described.

The rest of the paper is organized as follows. In the next section, related work on scalability and summarization is reviewed. Some basic concepts and the proposed framework for video summarization are introduced in Section III, while Sections IV, V and VI describe, respectively, the analysis, ranking and generation processes. The results of objective and subjective experiments are provided in Section VII. Finally, Section VIII draws the conclusions.

## II. Related Work

Video summarization has been addressed by many researchers with multiple approaches. However, most methods follow a single scale approach, that is, the output is always a single summary. Realizing that sometimes a single scale may be insufficient, hierarchical summarization approaches[3], [4], [5] exploit the narrative structure of video sequences to provide the users with a set of summaries with different levels of detail, according with narrative hierarchy (e.g. chapters, scenes, shots, frames) or cluster hierarchy[5]. Each level of this hierarchy is in fact a different scale, with summaries with increasing length across the scales, although the summaries are not scalable within each level. These scales provide a coarse grain scalability, which is exploited in hierarchical browsing applications, where different levels of detail can be selected in those parts the user is more interested in.

In general, we use the term scalable summaries for the case of summaries the length of which can be adjusted with some accuracy without running again the summarization algorithm. Similar to scalable coding, the objective is to process the sequence once and generate different summaries depending on the length constraints (analyze once, generate many). Although hierarchical summarization creates scalable summaries,

This work is partially supported by the Spanish Government under project TEC2007-65400 (SemanticVideo).

The authors are with the Video Processing and Understanding Lab, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 28049 Madrid, Spain.

it targets hierarchical browsing and summaries with few scales corresponding to different narrative structures, while in general scalable summarization could target a larger number of scales to adapt the summary in constrained situations in which the length of the summary must be limited to a specific value. [4] introduces the idea of scalable summaries in a hierarchical summarization system.

Apart from those based on hierarchical approaches, very few techniques create scalable descriptions of summaries. [6] describes a representation of video sequences based on a priority curve. When this curve is computed, a summary of any desired length can be easily created. However, the main disadvantage of this method is that it needs a prior manual annotation stage of the sequence.

In recent years, the development of scalable coding formats has enabled other synergies between the concept of scalability and video summarization, not related with scalable summarization. The processing of compressed domain data and temporal scalable coding structures facilitate the efficient generation of the bitstream of the summary using bitstream extraction[7]. The same approach can be combined with other scalabilities (e.g. spatial, quality) for the adaptation of the summaries to a specific usage environment (e.g. resolution, network capabilities)[8].

## III. Proposed Approach for Scalable Summarization

### A. Properties and Modalities of Video Summaries

Summaries are compact representations of a given content that can be visualized in a much lower amount of time than the content itself. A good summary should have two properties: *semantic coverage* and *visual pleasantness*. The first property means that the summary should preserve as much representative information as possible while discarding as much redundant information as possible, so its length can be reduced. The second property means that a summary should be not only informative but also comfortable and pleasant when it is visualized by a user. A summary is not useful if the information is shown too fast or the summarization process introduces annoying artifacts (e.g. unconnected bits of shot changes, excessive speeding up of the frame rate). In that case, the user will not be able to retain almost any of the semantic information conveyed by the summary. A better approach would be a less informative summary, but easier to view for the user.

Our approach deals with two specific, but widely used modalities of video abstracts. A *storyboard* is an abstract composed by selecting few independent and separated frames to represent the content in few images. In a storyboard there is no playback, as it is just a set of individual images. In contrast, a *video skim* has the form of a short sequence obtained by selecting certain segments of the input sequence. For storyboards, it is usually enough to focus on optimizing the semantic coverage. The only unpleasant effects come from frames belonging to transitions (e.g. fades, wipes, dissolves) as they contain mixed and incomplete information from two shots, so it is better to avoid including them in the storyboard. However, the editing operation involved in video skims may lead to artifacts, so both semantic coverage and visual pleasantness must be balanced in order to obtain informative summaries while avoiding annoying artifacts. Short segments belonging to shot changes are examples of unsuitable segments for video skims.

Obviously, the length of the summary plays an essential role in summarization. It must be significantly reduced but the summary must preserve as much information as possible. As described before, the length also affects not only the semantic coverage but also the visual aspect of the summary, especially for video skims. If the skim is very short but trying to include too much information, it will become annoying and quite probably useless. Most algorithms are designed for specific lengths, but in some cases the performance might be degraded when the required length is not in the expected range. However, depending on the application, it may be useful to have summaries with adjustable length.

### B. Embedded Summarization

An important assumption in our approach is that summaries can be generated very efficiently using bitstream extraction[8]. Bitstream extraction[9] has been extensively used together with scalable coding for fast bitstream adaptation. The same concept is used here in the context of video summarization. Since most modalities of summaries are built by the concatenation of frames taken from the source sequence, it is possible to assume that the summaries are embedded in the source sequence. The former is strictly true only if frames are coded independently (e.g. YUV format). The frame is the basic unit for summarization and the summary is just a set of indices of frames. However, when dealing with compressed sequences (we consider conventional I, P and B frames such as those used in MPEG-1, MPEG-2 and MPEG-4) the former is not always possible, as frames are related by the coding structure, and predicted frames cannot be decoded if their reference frames were discarded. For this reason, it is more suitable to use a slightly different model to describe the summaries, using coding units as basic units for summarization.

We use term *summarization unit* for a set of consecutive frames related by some coding structure, and that can be decoded independently of the other frames in the sequence. For convenience we use the Group of Pictures (GOP) $g_m$ as the main summarization unit in this model. Thus, the source sequence $V$ is coded in $M$ GOPs. Let $f_m$ denote the I frame belonging to the GoP $g_m$. This frame can be decoded independently of the other frames of the sequence, so it constitutes another summarization unit. Keyframe based summaries, such as storyboards, are composed of a set of I frames. Segment based summaries, such as video skims, are composed by joining complete GOPs. In this context, an *embedded summary* $S \subseteq V$ is a sequence of arbitrary summarization units (see Fig. 1a). The bitstream of the embedded summary $S$ is obtained from the input sequence $V$ using the *extraction* operation, which combines the summarization units into a valid bitstream.

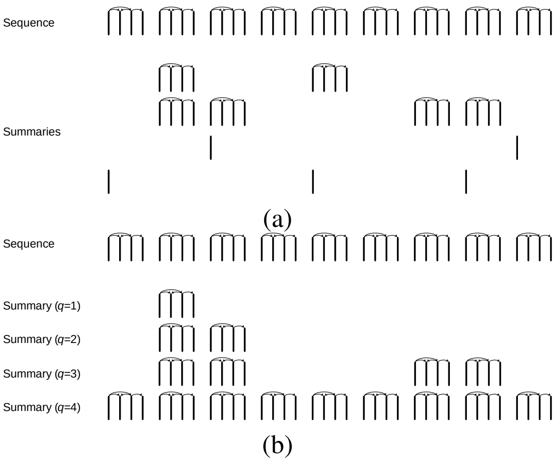


Figure 1. Models for video summaries: a) embedded , b) scalable.

### C. Scalable Summaries and Ranked Lists

Different lengths can be addressed using different embedded summaries. However, the concept of scalability can be also used. Thus, we introduce scalable summarization as a special case of embedded summarization (see Fig. 1b) with an important additional restriction. A *scalable summary* is a set of embedded summaries $SS = \left\{S^1, \cdots, S^q, \cdots, S^Q\right\}$, with $q \in \mathbb{N}$ denoting the summarization scale and $Q$ denoting the number of scales, and with each embedded summary $S^q$ satisfying

$$S^1 \subset S^2 \subset \cdots S^q \subset \cdots \subset S^Q \subseteq V \qquad (1)$$

In embedded summarization, summaries are described by a set of summarization units (GOPs in this paper). However, for scalable summarization we introduce the ranked list as a different tool to describe the whole set of summaries. A *ranked list* $\texttt{list}_{\texttt{SS}}$ is a sequence with the indices of the GOPs sorted by their relevance for summarization representing the scalable summary $SS$. A summary of length $M'$ GOPs, embedded in $SS$ contains the first GOPs with the first $M'$ indices in $\texttt{list}_{\texttt{SS}}$. Note that for each embedded summary a different set must be specified, but for scalable summaries, the ranked list contains all the summaries in a single compact representation. Thus, in scalable summarization, the objective of the analysis algorithm is to determine the ranked list.

### D. Overview of the Framework

As in scalable coding approaches, the analysis stage is detached from the generation/adaptation stage (see Fig. 2). While analysis is performed only once and its results are stored, generation is performed for each summarization request. For this reason, the generation stage must be very efficient, in order to fully benefit from the summarization scalability. The proposed framework is based on GOPs as basic units and extraction for efficient generation of the bitstream. The results of the analysis stage are stored in a ranked list. For each request and according to the required length, the generation module reads the ranked list of the sequence and determines which GOPs must be included in the summary. Then, the bitstream extractor processes the bitstream of the sequence to generate the bitstream of the summary.

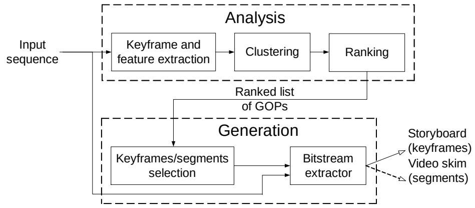


Figure 2. Proposed architecture

## IV. Analysis

The analysis module parses the input bitstream and structures the sequence into suitable units. The subsequent ranking algorithm processes these units to obtain the ranked list.

### A. Analysis Structure and Feature Extraction

The analysis is also based on GOPs as basic units. We assume that the original sequence has $M$ GOPs. Each GOP $g_m$ has one I frame $f_m$, which is used for feature extraction in order to characterize the GOP $g_m$. First, shot boundaries are detected both to structure the sequence into shots and to discard GOPs with transitions, which may lead to unpleasant effects in the summary. In this framework, GOP precision is usually enough rather than frame precision. For this reason we implemented a simple cut detector based on the thresholding approach described in [10], but applied in a GOP basis by processing the DC image[11] of each I frame $f_m$. GOPs containing shot changes and short shots (too few GOPs in length) are discarded for subsequent analysis to avoid potential artifacts in the summary.

Each valid shot is represented with few GOPs (selected by uniform sampling up to a maximum of $W_{max}$ keyframes per shot), and their I frames are used as keyframes. For each keyframe a feature vector is then computed. At this point, the sequence is partitioned into $R$ valid shots $s_r$, which are further represented by $P$ keyframes $t_p$. These keyframes will be clustered into $K$ clusters $c_k$.

### B. Clustering

A clustering algorithm is used to remove semantic redundancies between the keyframes, grouping those keyframes with similar features into the same cluster. In our approach, feature vectors are based on the MPEG-7 colour layout, along with a suitable distance[12], although other features or temporal information might be used for clustering[13]. The output of the clustering stage is a set of $K$ clusters $\{c_0, c_1, \ldots, c_{K-1}\}$. In principle, any clustering algorithm based on a distance $d\left(t_p, t_{p'}\right)$ between the feature vectors of every pair of keyframes $t_p$ and $t_{p'}$, could be used to group the keyframes into clusters. Conventional algorithms such as K-means, hierarchical clustering and spectral clustering have been used previously for video summarization. We use the hierarchical clustering algorithm with average linkage[14]. The number of clusters is determined using a threshold in the linkage distance. Finally, for each cluster $c_k$, the keyframe with its feature vector closest to the centroid is selected as the representative keyframe $\overline{t_k} \in c_k$.

Note that we do not use the hierarchy generated by the hierarchical algorithm for scalability. Instead of that we use a different approach based on iterative ranking to generate the scalable representation.

## V. Iterative Ranking

The ranking stage is motivated by the need to address the problem of generating suitable summaries for a wide range of potential lengths, but created in a single process. The objective is to obtain a scalable representation of storyboards and video skims. It follows an incremental growing approach, returning two ranked lists: $\mathtt{list_{sb}}$ for storyboards and $\mathtt{list_{vs}}$ for skims. Note that, in contrast to conventional approaches, this approach creates a set (with a high number of embedded summaries). The objective is to find a good set of embedded summaries satisfying (1) and not a single optimal summary.

### *A. Cluster Level*

When the length of the summary is very constrained (usually in storyboards and short skims), the algorithm focuses on covering the basic semantics of the sequence with as few GoPs as possible. We assume that clusters are a reasonably good representation of these semantics, and that each cluster can be represented by a keyframe (for storyboards) and by a short excerpt of $N_{exc}$ consecutive GOPs (for video skims). Thus, finding a representative keyframe or segment for each cluster should be enough to have a suitable summary in these limited conditions. If the length of the summary is even more constrained some keyframes or excerpts must be discarded, with an associated loss in coverage.

To provide the best set of keyframes for each summary length the clusters are ordered by their relevance, using the following iterative ranking procedure in which the clusters are ranked and selected incrementally:

1) Set scale $q = 0$. Set $S^q = \mathtt{list_{sb}} = \mathtt{list_{vs}} = \emptyset$.
2) Compute the score $H^q(c_k)$ for every cluster $c_k$. Select the cluster $c^*$ with maximum score. Mark $c^*$ as selected including it in $S^q$. Grow previous summaries as follows
    a) Include $m^*$ in $\mathtt{list_{sb}}$ , where $m^*$ is the index of the GOP with $\overline{t_k}$ (keyframe representative of $c^*$).
    b) Include an excerpt $\boldsymbol{b}$ in $\mathtt{list_{vs}}$ centered at GOP index $m^*$. The excerpt of fixed length $N_{exc}$ GOPs (with $N_{exc}$ even) is defined as
    $$\boldsymbol{b} = \left\{ m^* - \frac{N_{exc}}{2}, \ldots, m^*, \ldots, m^* + \frac{N_{exc}}{2} - 1 \right\}$$
3) Set $q = q+1$. Set $S^q = S^{q-1}$ . Go to step 2 and repeat until all the clusters are selected.

The score at the scale $q$ of each cluster is then given by

$$H^q(c_k) = \begin{cases} (1-\alpha_c)\frac{H_{dist}(c_k)}{\max\limits_j H_{dist}(c_j)} + \alpha_c \frac{H_{dur}(c_k)}{\max\limits_j H_{dur}(c_j)} & k \notin S^{q-1} \\ 0 & k \in S^{q-1} \end{cases} \tag{2}$$

Note that selected clusteres are no longer considered in the subsequent iterations. Note also that scores must be recalculated for each iteration, as there is not a global *a priori* score for each cluster (as in other summarization algorithms[1], [6]), but a local *a posteriori* score for each iteration, conditioned by the summary obtained in the previous iteration.

The scores are computed based on two criteria: distance and duration, combined in a weighted sum. The duration score favours the selection of clusters with more contribution in terms of duration of the sequence, assuming that longer clusters should be included at lower scales. The score $H^q_{dur}(c_k)$ is computed as

$$H^q_{dur}(c_k) = L(c_k) \tag{3}$$

where the duration of $c_k$ is defined as $L(c_k) = \sum\limits_p L(s_p)$, $\forall s_p \in c_k$, and $L(s_p)$ is the length of $s_p$ in number of GOPs. A shot $s_p$ belongs to $c_k$ if any of its representative keyframes is member of $c_k$. The distance scores favours the selection of clusters more dissimilar to those already selected in previous iterations. The distance $d(c_i, c_j)$ between clusters is computed as the distance between their representative keyframes (i.e., $d(c_i, c_j) = d(\overline{t_i}, \overline{t_j})$). The score $H^q_{dist}(c_k)$ is computed as

$$H^q_{dist}(c_k) = \begin{cases} 0 & q = 0 \text{ or } k \in S^{q-1} \\ \min\limits_{j \in S^{q-1}} d(c_k, c_j) & q > 0,\ k \notin S^{q-1} \end{cases} \tag{4}$$

Fig. 3 shows an example of the different scales in a scalable storyboard. As it can be observed, shorter summaries are included into longer ones. Also, clusters with long shots (head shots of interviewees) are included at earlier stages, due to the contribution of the duration score to the overall score.

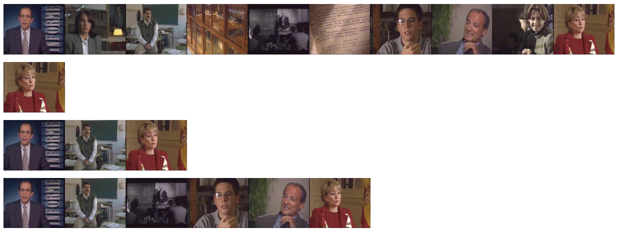


Figure 3. Example of scales of a scalable storyboard of *news11* ($\alpha_c = 0.5$).

### *B. Shot Level*

For longer skims, once a minimum semantic coverage is achieved and the length of the summary increases, the summaries can be improved not only from the semantic coverage point of view, but also emphasizing other aspects. In this sense, segments with different lengths can be allowed in the summary, becoming the summary more natural. At this level, the ranking is performed iteratively for each GOP computing scores for each shot. For each iteration, there are two possible actions: including an additional excerpt from a new shot or growing an excerpt included previously. The ranking algorithm continues after cluster ranking with the following steps:

1) Compute $score^{q}\left(s_r\right)$ for each shot $s_r$. Select the shot $s*$ with maximum score. Grow summaries:
   a) If $s*$ was not selected previously, select the keyframe $t*$ closer to the middle of the shot $s*$ and include an excerpt $\boldsymbol{b}$ centered at the GOP index of $t*$ in $\texttt{list}_{\texttt{vs}}$. Mark $s*$ as selected including it in $S^{q}$.
   b) If $s*$ was already selected, grow the selected excerpt with an additional GoP of index $m*$ (alternatively from left and from right bounds of the excerpt until shot bounds are found). Update $\texttt{list}_{\texttt{vs}}$ including $m*$.
2) Set $q = q + 1$. Set $S^{q} = S^{q-1}$ . Go to step 2 and repeat until all the GOPs in all shots are selected.

As in cluster ranking, the score $scr^{q}\left(s_r\right)$ weights two different criteria with $\alpha_s$ controlling the trade-off.

$$H^{q}\left(s_r\right) = \begin{cases} \left(1-\alpha_s\right)\frac{H^{q}_{dist}(s_r)}{\max\limits_{j} H^{q}_{dist}(s_j)} + \alpha_s \frac{H^{q}_{dur}(s_r)}{\max\limits_{j} H^{q}_{dur}(s_j)} & r \notin S^{q-1} \\ 0 & r \in S^{q-1} \end{cases} \tag{5}$$

The score $H^{q}_{dist}\left(s_r\right)$ favours the inclusion of shots which are not well represented by the current summary, and it is based on the Hausdorff distance from the unselected keyframes ($\tilde{t}^{q}$ represents the set of selected keyframes at scale $q$) of the shot $s_r$ to the current summary. If all the keyframes belonging to the shot $s_r$ were included previously, then the score is zero. Thus, the score $H^{q}_{dist}\left(s_r\right)$ is calculated as

$$H^{q}_{dist}\left(s_r\right) = \begin{cases} \max\limits_{\substack{t_j \in s_r \\ \forall t_j \notin \tilde{t}^{q-1}}} \left\{ \min\limits_{\forall t_j \in \tilde{t}^{q-1}} d\left(t_j, t_k\right) \right\} & if \begin{array}{c} \exists t_j \in s_r | \\ t_j \notin \tilde{t}^{q-1} \end{array} \\ 0 & otherwise \end{cases} \tag{6}$$

The score $H^{q}_{dur}\left(s_r\right)$ is calculated as

$$H^{q}_{dur}\left(s_r\right) = e^{-\left(\frac{\rho(s_r)}{\max\limits_{t_j, t_k \in s_r} d(t_j, t_k)}\right)^{\lambda}} \tag{7}$$

The duration criterium is based on the assumption that longer shots should be represented with longer excerpts in the summary. This ad-hoc expression measures the relevance of the shot according to $\rho\left(s_r\right) = 1 - \frac{L\left(s_r \bigcap \texttt{list}^{(q)}_{\texttt{vs}}\right)}{L(s_r)}$, which is the ratio between the duration of the shot not yet selected at the scale $q$ and the total duration of the shot (in GOPs). To avoid an excessive prominence of longer shots, we use an exponential function (in the experiments we used $\lambda = 4$). When $\rho\left(s_r\right)$ reaches a sufficient value, the shot is considered well represented and $H^{q}_{dur}\left(s_r\right)$ decreases rapidly.

Finally, the GOPs initially discarded in the analysis stage can be included in the ranked list if a complete scalable representation of the sequence is required.

## VI. Generation

Once the analysis stage has obtained the scalable summary and has coded it as a ranked list $\texttt{list}$, any of the embedded summaries can be extracted from the bitstream of the original sequence. The extracted summary will depend on the constraints imposed by the user or the context (e.g. number of images, duration of the skim). For example, given the length of the skim $L_{skim}\left(q\right)$ and a maximum length $L_{max}$, the cutoff index in the ranked list is

$$q_{max} = \underset{\substack{L_{skim}(q) \leq L_{max} \\ 1 \leq q \leq |\texttt{list}|}}{\arg\max} \; L_{skim}\left(q\right) \tag{8}$$

The set with the indices of the summary is obtained as the first $q_{max}$ indices in $\texttt{list}$. The actual bitstream of the summary is then efficiently generated using bitstream extraction, which consists of the selection of the adequate packets from the input bitstream.

## VII. Experimental Results

### A. Shot Change Detection

We first evaluated the shot change detection algorithm with the sequence *NASASF-TheTechnicalKnockout* from the TRECVid 2005 shot boundary detection corpus. This sequence has 105660 frames that were encoded in MPEG-2 using diferent GOP lengths. The results (see Table I) show a reasonably good performance for all the GOP lengths studied in the experiment. However, for longer sizes, GOP processing may not be enough for an effective detection of cuts, and the decoding of the rest of frames (or DC images) would be necessary.

### B. Subjective Evaluations

We conducted an experiment designed to evaluate summaries at several scales. Quality assesment of summaries becomes even more complex in this case, as every summary must be evaluated at a number of scales. The evaluation data set consisted of three segments of 10 minutes of the sequences *contesting* and *news11* (from MPEG-7 Content Set) and *BBC rushes* (from TRECVid 2007, specifically MRS042538). They cover the range between the highly structured and low redundant news content (*news11*) to the highly redundant and repetitve content of unedited footage (*BBC rushes*). We used a GOP size of 8 frames.

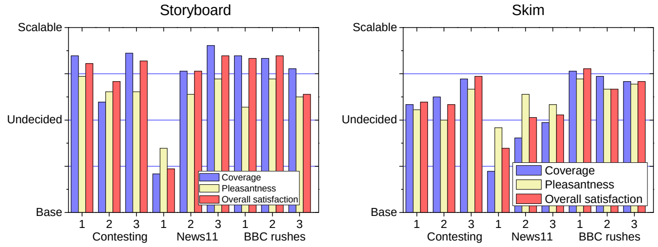
We compared the scalable method (with $W_{max} = 3$, $\alpha_c = 0.1$, $\alpha_s = 0.1$ and $N_{exc} = 4$) with a baseline method consisting of a uniform sampling of the sequence either in frames for storyboards or short segments for video skims. Although the method is very simple, it is easy to replicate and widely used in video libraries. Note that this method does not generate scalable summaries as they were defined previously. In a preliminary evaluation of this baseline approach we found that the assessors were satisfied with the summaries, especially for structured content such as news or documentaries.

We evaluated three scales for each test sequence, for both storyboards and skims, comparing the baseline and the scalable versions of each summary. A total of 18 volunteers were asked to assess which summary consider better according to three evaluation criteria: information coverage, visual pleasantness and overall satisfaction.

Table I
RESULTS OF SHOT CHANGE DETECTION EXPERIMENT. FP: FALSE POSITIVES, M: MISSED, R: RECALL, P: PRECISION.

| Sequence | | Abrupt | | | | | Gradual | Total | | |
|---|---|---|---|---|---|---|---|---|---|---|
| GOP length | #GOPs | # | FP | M | R | P | # | # | R | P |
| 1 | 105660 | 604 | 33 | 35 | 0.942 | 0.945 | 95 | 699 | 0.814 | 0.945 |
| 2 | 52829 | 607 | 45 | 36 | 0.940 | 0.926 | 92 | 699 | 0.816 | 0.926 |
| 4 | 26414 | 622 | 34 | 47 | 0.924 | 0.944 | 77 | 699 | 0.822 | 0.944 |
| 8 | 13206 | 629 | 36 | 52 | 0.917 | 0.941 | 68 | 697 | 0.827 | 0.941 |
| 16 | 6602 | 660 | 48 | 99 | 0.850 | 0.921 | 33 | 693 | 0.809 | 0.921 |

The results of the evaluations (see Fig. 4, where numbers 1, 2 and 3 indicate the scale) confirm a preference, in general, for the scalable method, except for the sequence *news11*, for which the baseline method is preferred at low scales. This preference is more evident for storyboards than for skims. For the sequence *news11* the results are different and the baseline method generates very good summaries for this kind of structured video, implicitly exploiting its regular pace and distribution of information.



(a)

(b)

Figure 4. Relative evaluations: a) storyboards, b) skims.

## C. Performance

Efficiency in the generation of the bitstream is crucial for scalable summarization. Table II shows the highly efficient analysis and generation in the case of the sequence *news11* (on an Intel Pentium M 1.86 Ghz processor), especially for typical summaries shorter than 10% of the original duration. For longer summaries, it shows an approximately linear trend.

Table II
PROCESSING TIME (IN SECONDS) FOR THE SEQUENCE *news11*.

| Analysis | Generation | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Storyboard (#) | | Skim (%) | | | | | |
| | 5 | 30 | 1 | 5 | 10 | 20 | 50 | 100 |
| 2.21 | 0.64 | 0.68 | 0.68 | 0.70 | 0.79 | 1.46 | 7.09 | 16.90 |

## VIII. CONCLUSIONS

In this paper, the concept of scalable summarization was discussed, along with the effect of the scale on different types of summaries. The problem is addressed with an incremental growing approach, in which the summary at each scale is obtained by improving the previous one. To implement this approach efficiently, a compressed domain framework was also proposed, with fast analysis and generation of storyboards and video skims. Fast generation of the bitstream is crucial to fully benefit from the advantages of scalable representations. Iterative ranking was proposed to tackle adaptively both short and long summaries, with fine granularity. Experimental evaluations showed reasonably good results at different scales and in different contexts.

The proposed approach is generic, focused on the framework itself and the scalable approach. However, even though the analysis relies on simple features, scores and analysis algorithms, the results obtained are promising. We think that the algorithm can be further improved, as the analysis stage can be enhanced with high level features and semantic analysis. Specific scoring methods for different contexts can also help to create better summaries.

## REFERENCES

[1] B. T. Truong and S. Venkatesh, “Video abstraction: A systematic review and classification,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 3, no. 1, p. 3, 2007.

[2] B. Tseng, C.-Y. Lin, and J. Smith, “Using mpeg-7 and mpeg-21 for personalizing video,” *IEEE Multimedia*, vol. 11, no. 1, pp. 42–52, 2004.

[3] J. Bescos, J. M. Martinez, L. Herranz, and F. Tiburzi, “Content-driven adaptation of on-line video,” *Signal Processing: Image Communication*, vol. 22, pp. 651–668, 2007.

[4] X. Q. Zhu, X. D. Wu, J. P. Fan, A. K. Elmagarmid, and W. F. Aref, “Exploring video content structure for hierarchical summarization,” *Multimedia Systems*, vol. 10, no. 2, pp. 98–115, Aug. 2004.

[5] S. Benini, A. Bianchetti, R. Leonardi, and P. Migliorati, “Extraction of significant video summaries by dendrogram analysis,” in *Proc. IEEE International Conference on Image Processing*, 2006, pp. 133–136.

[6] M. Albanese, M. Fayzullin, A. Picariello, and V. Subrahmanian, “The priority curve algorithm for video summarization,” *Information Systems*, vol. 31, no. 7, pp. 679–695, Nov. 2006.

[7] L. Herranz and J. M. Martínez, “On the use of hierarchical prediction structures for efficient summary generation of H.264/AVC bitstreams,” *Signal Processing: Image Communication*, vol. 24, no. 8, pp. 615 – 629, 2009.

[8] ——, “An integrated approach to summarization and adaptation using H.264/MPEG-4 SVC,” *Signal Processing: Image Communication*, vol. 24, no. 6, pp. 499–509, 2009.

[9] G. Panis, A. Hutter, J. Heuer, H. Hellwagner, H. Kosch, C. Timmerer, S. Devillers, and M. Amielh, “Bitstream syntax description: a tool for multimedia resource adaptation within MPEG-21,” *Signal Processing: Image Communication*, vol. 18, no. 8, pp. 721–747, Sep. 2003.

[10] Y. Nakajima, K. Ujihara, and A. Yoneyama, “Shot change detection from partially decoded MPEG data,” *Systems and Computers in Japan*, vol. 30, no. 8, pp. 11–22, 1999.

[11] B.-L. Yeo and B. Liu, “On the extraction of DC sequence from MPEG compressed video,” in *Proc. International Conference on Image Processing*, vol. 2, 1995, pp. 260–263 vol.2.

[12] E. Kasutani and A. Yamada, “The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval,” in *Proc. International Conference on Image Processing*, A. Yamada, Ed., vol. 1, 2001, pp. 674–677 vol.1.

[13] H. Yi, D. Rajan, and L.-T. Chia, “A motion-based scene tree for browsing and retrieval of compressed videos,” *Inf. Syst.*, vol. 31, no. 7, pp. 638–658, 2006.

[14] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 2006.