Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Scalable storyboards in handheld devices: applications and evaluation metrics

Luis Herranz · Shuqiang Jiang

Received: date / Accepted: date

Abstract Summaries are an essential component of video retrieval and browsing systems. Most research in video summarization has focused on content analysis to obtain a compact yet comprehensive representation of video items, while some important related aspects such as how they can be effectively integrated in mobile interfaces and how to predict the quality and usability of the summaries. Conventional summaries are limited to a single instance with certain length (i.e. a single scale). In contrast, scalable summaries target representations with multiple scales, that is, a set of summaries with increasing length in which longer summaries include more information about the video. Thus, scalability provides high flexibility that can be exploited in devices such as smartphones or tablets to provide versions of the summary adapted to the limited visualization area (which also varies from device to device). In this paper, we focus on scalable storyboards and explore their application to summary adaptation and zoomable video navigation in handheld devices. By introducing a new adaptation dimension related with the summarization scale, we can formulate navigation and adaptation in a two-dimensional adaptation space, where different navigation actions modify the trajectory in that space. We also describe the challenges to evaluate scalable summaries and some usability issues that arise from having multiple scales, and propose some objective metrics that can provide useful insight about the potential quality and usability without requiring very costly user studies. Experimental results show a reasonable agreement with the trends shown in subjective evaluations. Experiments also show that content-based scalable storyboards are less redundant and useful than the content-blind baselines.

L. Herranz · S. Jiang
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology
Chinese Academy of Sciences, Beijing 100190, China

L. Herranz
E-mail: luis.herranz@vipl.ict.ac.cn
S. Jiang
E-mail: sqjiang@ict.ac.cn

The final publication is available at Springer via http://dx.doi.org/10.1007/s11042-014-2421-4

1 Introduction

Visual surrogates, and in particular video summaries, are an essential component of video retrieval and browsing applications, especially when the particular content is long and complex (e.g. movies, news programs, documentaries). A search query in YouTube often returns a large number of results, each of which may be also long and with complex temporal structure. Similarly, a folder or a personal collection may contain hundreds of videos. Summaries can save time and effort by providing compact representations preserving the essence of the content.

Mobile networks, smart handheld devices and inexpensive storage space have played a crucial role in the explosion of video content, as not only professionals but also common users now can contribute with their own content. In particular, smartphones and tablets have changed the traditional way to interact with multimedia content, and particularly with video. These devices integrate high quality recording, so any user can record videos and share them from any place and in any situation. Nowadays, enjoying a movie on the bus, watching the latest news or sharing your experiences during a trip are common situations enabled by these technologies. . However, dealing with this huge amount of data is particularly challenging in handheld devices due to the limited size of the screens.

Finding interesting content in a large list of videos is inherently multiscale. As we often are just interested in a short segment (e.g. sports news in a news bulletin, a specific scene of a movie, a funny moment in a TV show), the problem is not only limited to deciding which video but also to finding the segment of interest. However, commonly used surrogates such as one keyframe, the title or a short textual description can hardly grasp the underlying visual content in the video and present it in an intuitive way. Visual summaries[40], such as storyboards, provide a more intuitive way to explore videos and video collections. However, these surrogates are longer, requiring more time be visualized. In cases when the display area is limited (e.g. smartphones), shorter summaries requiring smaller area are also preferred. Motivated by this problem, we focus on *scalable storyboards*[18,8], in which summaries of different lengths can be obtained with fine granularity. Thus, scalable storyboards can effectively balance summary length and amount of information depending on the particular needs, with potential applications in browsing and adaptation.

While many methods can generate, implicitly or explicitly, scalable storyboards, most of them just ignore this aspect. Thus, the implications of summarization scalability are not studied. In that sense, models to represent scalable summaries and their potential applications have been barely explored. Another important problem is how to evaluate the quality of a summary and particuarly a multiscale summary. An exhaustive subjective evaluation of scalable summaries is very challenging due to a dramatic increase in the number of instances to be evaluated. There are no specific evaluation protocols for scalable summaries, and, in particular, no suitable objective quality metrics.

In this paper we address some of these unexplored issues and describe potential applications in the context of handheld devices (e.g. smartphones, tablets). In our model, different instantiations of a scalable summary are represented as points in a multidimensional space, and adaptation consists in finding the optimal instance in a constrained optimization problem. Then we show how this model can be applied in content adaptation and multiscale summary navigation. In the latter, some usability aspects must be also considered. In particular, smooth transitions between different

scales of the summary can be even more important than the summary itself. In addition we highlight the challenges of evaluating the quality of scalable storyboards, and propose some objective measures that we can use to compare different methods. A user study reveals certain correlations between these objective measures and the subjective measures.

The rest of the paper is organized as follows. Section 2 review previous works in related areas. The framework of multiscale storyboards and some basic concepts are introduced in Section 3, and several specific algorithms to generate them are described in Section 4. In Section 5, the adaption model is described together with some applications. The problem of evaluating multiscale summaries is discussed in Section 6, where some objective metrics are also proposed. Finally, Section 7 and Section 8 describe the experiments and present the conclusions, respectively.

2 Related Work

2.1 Video summarization and adaptation

Visualizing video content is time-consuming due to its intrinsic temporal nature. So in order to provide effective and fast browsing, surrogates can provide a quick idea about the content in just a fraction of the original duration. Compared with textual surrogates such as titles and descriptions, visual abstracts (e.g. thumbnails, short clips) are much more intuitive and suitable to browse video[24]. However, most video browsing and retrieval systems still use a single thumbnail, which is usually insufficient to grasp all the complexity of a long video. The most important visual surrogates are storyboards and video skims. The former is a very compact representation consisting of a sequence of still images (keyframes) extracted from the source sequence. The latter consists of a short sequence built with excerpts of the source sequence, less compact representation but preserving the dynamic nature of video.

Video summarization is a challenging field as it involves high level understanding to select representative pieces of the original video, and present them in an effective, intuitive and appealing way[40,26]. Unfortunately, the so called *semantic gap*[39] is still very large, and complex methods often just contribute with very minor improvements compared with simple content-blind baselines (e.g. evenly spaced keyframes, fast-forward)[30], or do not make any difference at all[32]. Object recognition[46] and event detection techniques[44,33] can help to guide the summarization of complex videos, by detecting and highlighting objects and events of interest. Subtitles and other textual cues provide useful information that can be leveraged to generate more abstract summaries[11,9]. Sometimes, preserving semantic information is not the most important aspect. For instance, affective analysis[48,47] can be included to create specialized summaries such as movie trailers[20], where the emotions often play a more important role than semantic information. For the purpose of this paper, we restrict our discussion to summaries where the objective is to preserve semantic information.

On the other hand, content adaptation[7] aims at maximizing the user experience by creating and delivering a suitable version of the content, adapted to the specific usage conditions (e.g. different terminals and networks)[42]. Particularly important is the case of handheld devices, where other issues such as limited computational resources and low power consumption requirements become very important. For instance, transcoding[2,

43] and scalable video coding[29, 1, 36] can deliver bitstreams with different resolutions, bitrates or frame rates, according to the particular requirements.

Often, the content itself can be analyzed and exploited to improve the adaptation. Thus, video summarization can be also considered as a special type of content-aware adaptation (or semantic adaptation), in which the structure of the video is modified to show only some key parts providing more a compact representations of the content[7]. Furthermore, video summaries themselves could be adapted to the usage context (e.g. the frame size and bitrate of a video skim can be also adapted to the terminal and network). Thus, summarization and adaptation can be often integrated in the same framework[17].

Adaptation is usually posed as an optimization problem with constraints where the objective is to maximize certain utility function (e.g. perceived quality, user experience). Standard tools to describe terminals and networks (e.g. screen resolution, network bitrate, decoding capabilities) and constraints and utility functions are included in the standards MPEG-7 and MPEG-21[42]. Scalable approaches are well suited for these type of problems, as adaptation consists in finding the adaptation coordinates, and the actual generation of the adapted bitstream is very simple and fast. For instance, scalable image and video adaptation are often formulated in this way[27]. Similarly, considering video summaries as intrinsically scalable data, in this paper we exploit the idea of scalability for summarization, navigation and adaptation.

2.2 Multiple scales in video summaries

Traditionally, research in video summarization has been focused on content analysis and redundancy removal, implicitly considering a single scale. As sometimes a single scale may be insufficient, hierarchical summarization approaches[49, 50, 4, 6] exploit the narrative structure of video sequences to provide the users with a set of summaries with different levels of detail, according to a narrative hierarchy (e.g. chapters, scenes, shots, frames). Each level in this hierarchy is in fact a different scale, and summaries increase their lengths as we include lower levels. However, these summaries are not scalable within each level. These scales provide a very coarse grained scalability with rigid boundaries related to the hierarchy, which is exploited in hierarchical browsing applications, but not so adequate for adaptation.

In contrast, scalable summarization should aim at a larger number of scales in order to address scenarios requiring fine adjustment of the summary length. Scalable summaries have a number of applications, ranging from the customized adaptation of video summaries to a given length and progressive video access, to visualization and interactive video browsing. Zhu et al[50] introduce some degree of scalability at the lowest level of a hierarchical summary. Some methods can implicitly create scalable summaries, although they do not exploit their multiscale nature. Albanese et al[3] describe 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 drawback of this method is that it needs manual annotation of the sequence. An iterative growing algorithm[16] has been proposed to generate scalable storyboards and video skims with fine granularity. This approach emphasizes scalability in video summaries, where length can be adjusted on demand at adaptation time, without the need of running again the entire summarization process. A similar approach is proposed in [25], which proposes a method to construct size-constrained storyboards, formulated in

terms of spanning trees. Based on the idea of semantic concept preservation, Yuan et al[45] also propose a similar ranking method to generate scalable storyboards. Scalability in more complex summary formats has been also explored, including video skims[18, 5,8] and comic-like summaries[15].

2.3 Evaluation of video summaries

An important pending issue is the evaluation of summaries (and particularly multiscale summaries), related with the lack of reliable objective measures[40]. The alternative is to involve human assessors in the evaluation process (e.g. user studies, annotation of subjective ground truth), which makes the evaluation considerably more complex, time-consuming and not easily replicable. It is not difficult to see that the complexity increases dramatically when multiple scales must be evaluated.

The most notable effort towards a common evaluation framework of video summaries is the TRECVID rushes evaluation[31,30]. Recently, several authors[10,21,41] even proposed methods to automatically estimate evaluation scores based on data from past campaigns. Unfortunately, rushes consist of highly redundant unedited footage, including *junk* segments (e.g. blank frames, test patterns, clapper sequences)[10], and rarely found in practical situations. Besides, particularly important in our case, summaries are only evaluated at one scale, corresponding to the target summarization rate (2%-4%, depending on the campaign[31,30]). Thus, the participants typically design the algorithms for that particular rate, but it is not clear how they would perform at other rates. All these reasons combined make comprehensive multiscale evaluation very rare. For instance, among all the multiscale methods reviewed in this section only few methods include some kind of multiscale evaluation (typically only three scales)[50,3, 4,6,18,5,15], while in the rest the evaluation is limited to one scale of the resulting summaries[25,45,8].

3 Multiscale storyboards

We separate the concept of multiscale storyboard, its description and the adaptation mechanism to extract a specific instance with a required length.

3.1 Overview

The framework has two stages (see in Fig. 1): analysis and adaptation. The former is performed only once (offline) and its objective is to analyze the content to extract the set of keyframes and obtain the scalable storyboard description (only required for content-based methods). This stage includes the sampling of the intial set of keyframes and the core of the summarization process itself. It could be performed either in the client or in a server. For computational reasons, we sample keyframes at fixed intervals, but other suitable keyframe sampling methods could be also used[23].

The second stage, i.e adaptation, is performed in the client each time a new storyboard is requested by the user interface. Each request has specific properties, depending on external constraints such as screen size or device orientation and the interaction of the user, who may change user interface parameters as a result of navigation. The

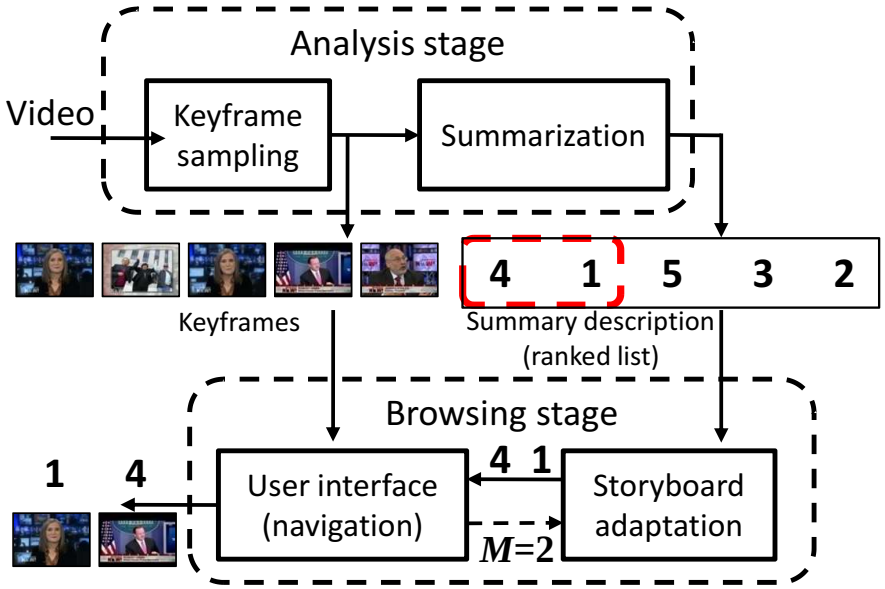

Fig. 1 Overview of the proposed system.

problem is solved in two steps: first, the user interface determines the adaptation coordinates, and determines the required number of images M. Then, the storyboard adaptation module will select, using the scalable description, an appropriate scale of the summary with length M. Thus, the user interface can compose a storyboard adapted to the specific requirements.

Note that the process is highly asymmetric, with most of the computational burden shifted to the offline analysis stage. In contrast, the on-demand adaptation process is lightweight and very fast, which is critical for usable and efficient user interfaces. The necessary analysis information to communicate both stages is stored in a suitable description format (i.e. ranked list).

3.2 Basic concepts

For convenience, we first define some preliminary concepts and the notation used in the rest of the paper (see Table 1).

By sampling uniformly the frames of the full video sequence, we obtain a set of keyframes $V = (I_1, \ldots, I_N)$, with I_n denoting the nth keyframe. This set V can range from the whole set of video frames to a smaller subset of frames.

We define a *storyboard* (implicitly referred to the set of keyframes V) as a subsequence

$$S = (I_{k_1}, \ldots, I_{k_m}, \ldots, I_{k_M} | k_m < k_{m+1}, \forall k_m, I_{k_m} \in V)$$

Thus, the storyboard is also determined by the sequence of indices $G = (k_1, \cdots, k_m, \cdots k_M)$, which is often more convenient. The length of S is $M = |S| \leq N$.

A *multiscale storyboard* SS is just a collection of storyboards with increasing lengths defined as

Symbol	Description
V	Keyframe set
S	Storyboard
I_k	kth image in the keyframe set V
k_m	Index of the mth image in the storyboard S
SS	Scalable storyboard
$S^{(q)}$	Scale q of a scalable storyboard SS
N	Number of images in the keyframe set V
M	Number images in a summary S
Q	Number of scales in a scalable summary SS
L	Ranked list representing a scalable summary SS

Table 1 Summary of notation.

$$SS = \left\{ S^{(1)}, \cdots, S^{(q)}, \cdots, S^{(Q)} | \left|S^{(1)}\right| < \cdots < \left|S^{(q)}\right| < \cdots < \left|S^{(Q)}\right| \right\}$$

where Q is the number of scales, $S^{(q)}$ is the summary at the scale $q = 1, \ldots, Q$ and $\left|S^{(q)}\right|$ is the length of $S^{(q)}$. We also assume that $S^{(Q)} = V$ corresponds to the whole set of keyframes. We can also define the step between two scales q and $q+1$ as the difference between the length of the summaries between the two scales $D(q+1, q) = \left|S^{(q+1)}\right| - \left|S^{(q)}\right|$. The step can vary, but in this paper we will focus on fine grained scalability with a constant step of one image, so the summaries are highly adaptable.

Particularly we will prefer a summary in which the transition is smooth in terms of information, that is, that most of the information of the previous scale is preserved and the new scale only adds a small amount of information. Thus, the user can easily track the new information. More formally, we say that a multiscale storyboard is *scalable* when $\left|S^{(q+1)} \setminus S^{(q)}\right| \ll \left|S^{(q+1)}\right|$where $\setminus$ is the set difference. We further say that the scalable storyboard is *strictly scalable* if $S^{(q+1)} \setminus S^{(q)} = \emptyset$, which means that all the information from the previous scale is preserved, and thus the amount of new information is minimal. In other words, it satisfies $S^{(1)} \subset S^{(2)} \subset \cdots \subset S^{(Q)}$.

3.3 Ranked list

There are many ways to describe the different summaries in a multiscale summary. When the size of the keyframe set and the number of scales are both high, the amount of information in the description also increases. For this reason, a compact representation is also desirable. Ranked lists were proposed[16] as a convenient and very compact representation of a highly scalable summary. A *ranked list* consists of the indices of the images in V, reordered by their relevance for summarization $L = (l_1, \ldots, l_i, \ldots, l_N | l_i \neq l_j, l_i \in \{1, \ldots, N\})$.

For a storyboard of length M, the corresponding images are those with the first M indices in the list, conveniently reordered by increasing value, in order to show them in temporal order. This description is only valid for strictly scalable summaries. However, we can easily extend the idea of ranked list to describe any multiscale summary. Implicitly, each new index l_i in the list is related with an insertion operation, i.e. insert the keyframe I_{l_i}. In the case of multiscale summaries, sometimes it may be necessary to remove keyframes and substitute them with others. So it is necessary a

complementary operation for deletions. The removal of a keyframe I_{l_i} is signalled with the sequence $(-1, l_i)$, since -1 is never used for indexing keyframes. Note that for (not strictly) scalable summaries, the number of deletions is still small, so the ranked list is still a very compact representation.

3.4 Storyboard adaptation

The list L is the only information required to recover any storyboard without any further processing, and the generation of the adapted storyboard is simple and fast. This is a lightweight process which does not need to process the video itself, just the ranked list and the previously stored keyframes. Thus, given the set of keyframes $V = (I_1, \ldots, I_n, \ldots, I_N)$ and an extended ranked list $L = (l_1, \ldots, l_{N'})$ with $N' \geq N$, to recover a storyboard S with length $M \leq N$ images we basically need to iterate over the list performing insertion and deletion operations until the required length is matched (see Algorithm 1).

Algorithm 1 Storyboard adaptation.

Input: length M; list L, keyframes V
Output: storyboard S

Set $p \leftarrow 1$,$m \leftarrow 1$, $G \leftarrow \emptyset$
while $m \leq M$ **do**
  Read l_p from L
  if $l_p \neq -1$ **then** *// Insertion*
    Append l_p to G
    Set $q \leftarrow q + 1$, $m \leftarrow m + 1$
  else *// Deletion*
    Read l_{p+1} from L
    Find $g = l_{p+1} \in G$, and remove it from G
    Set $q \leftarrow q + 2$, $m \leftarrow m - 1$
  end if
end while
Sort the elements in G in increasing order
Compose $S = (I_{g_1}, \ldots, I_{g_M} | \forall g_m \in G, I_{g_m} \in \mathrm{V})$
return S

4 Summarization methods

There are different ways to obtain storyboards with different lengths (i.e. to select a subset of keyframes from V). We can distinguish between content-blind and content-based methods. The former do not require any content analysis, and the algorithm is basically a deterministic rule to select images at specific temporal positions, regardless of their visual content. They are implicitly used in many practical systems because they can generate useful summaries with almost no cost. However, they have limitations as we will see later. Content-based methods first analyze the content trying to come up with a better way to select images for each specific video item. Consequently, a specific description is required for each item.

In the experiments we will compare five methods: temporal order, uniform sampling, iterative ranking and two clustering-based methods (see Table 2). The last two

Method		Content based	Type of scalability	Description
Deterministic	Temporal order	No	Strictly scalable	-
	Uniform sampling	No	Multiscale	-
Clustering-based	Hierarchical	Yes	Scalable	Ranked list
	K-means	Yes	Multiscale	Ranked list
Incremental growing	Iterative ranking	Yes	Strictly scalable	Ranked list

Table 2 Summarization methods compared in this study.

methods are content-based, exploiting visual similarity between keyframes to avoid non informative and redundant images.

4.1 Temporal order

In this method, if there is only space for $M \leq N$ images, then the first M images in V are selected, i.e., $S = (I_1, \ldots, I_M)$. The main drawback is that the storyboard does not cover the whole sequence, only its beginning. This is the simplest method, just included as a baseline reference, as it is used implicitly in most browsing applications when the whole storyboard does not fit into the available area.

4.2 Uniform sampling

Consecutive keyframes are often sampled from the same or similar scenes. In absence of any other knowledge about the video, it seems to be a better idea to distribute the selected keyframes evenly along the sequence. Thus, a better temporal coverage is achieved, being more likely to select content from different parts of the sequence (e.g. scenes, news sections). Not requiring analysis of the content and achieving a resonable coverage, this method is used implicitly in many browsing systems for handheld devices[19]. However, it still does not prevent from selecting redundant keyframes. In this case, a subset of M images is selected by resampling V at approximately equidistant intervals. In particular, we use

$$G = \left(\left\lfloor \frac{N}{M} \left(k_m - \frac{1}{2} \right) \right\rfloor + 1 | k_m < k_{m+1}, m = 1, \ldots, M \right) \quad (1)$$

and the corresponding summary is $S = (I_{k_1}, \ldots, I_{k_M} | k \in G)$.

4.3 Iterative ranking

The previous methods ignore the content itself, but good abstractions are necessarily content-dependent. The iterative ranking method[16,18] tries to iteratively add a new image to the summary, which is selected from the set of remaining unselected images (or video segments for video skims) according to a ranking function which is computed depending on the images selected in the previous iteration. It uses two criteria to select the most suitable image, based on its representativeness in terms of duration (the total duration of the shots represented by the image), and based on the visual distance to the previously selected images. In our case, we only consider a simplified version based only

on visual distance, since we do not perform shot detection. Thus, this simplification is equivalent to the method proposed in [25], which formulates the same approach in terms of spanning trees.

Algorithm 2 Iterative ranking-based list generation.

Input: set of keyframes V, number of keyframes N
Output: ranked list L

```
    // Initial summary
Compute g* = rep(V)
L ← g*
    // Summary growing
for q ← 2 : N do
    Compute score(I_x; S^(q−1)) for all I_x ∈ V
    Find g* = arg max_{x, I_x ∈ V} score(I_x; S^(q−1))
    Append g* to L
end for
    return L
```

The method is described in Algorithm 3. It starts with one image and iteratively increments the length of the summary by including the image which is at the highest visual distance from the set of already selected images. The idea is that by selecting the least similar image to those in the previous summary, we maximize the differential information with the previous scale. A score based on a point-to-set distance is computed as

$$score\left(I_x; S'\right) = \begin{cases} \min_{I_y \in S'} d\left(I_x, I_y\right) & I_x \notin S' \\ 0 & otherwise \end{cases} \tag{2}$$

where I_x and I_y are keyframes in V, S' is the summary in a previous scale (typically $S^{(q-1)}$), and $d\left(x, y\right)$ is the distance between them.

Note that the process is incremental, with one new image included in the summary after each iteration and no keyframe is removed. Thus, this method generates strictly scalable storyboards.

4.4 Clustering

A common approach in summarization is clustering[51, 14, 12, 28, 40]. Images are grouped into a few clusters $\{C_1, \cdots, C_K\}$, each of them $C_k = \{I_{k_1}, \cdots, I_{k_P}\}$ containing similar images. Then each cluster is represented by a single image. Thus, the whole video is represented by a few images presented as a storyboard. Determining the number of clusters is usually difficult, and it may also depend on the application and preferences. Here we address multiscale summaries, so we compute scales with different numbers of clusters.

Let $\mathcal{H} = \left\{\left\{C_1^{(1)}, \cdots, C_{N_1}^{(1)}\right\}, \cdots, \left\{C_1^{(Q)}, \cdots, C_{N_Q}^{(Q)}\right\}\right\}$ denote the set of clusterings, where $C_i^{(q)}$ is the ith cluster at level q. We assume that they have increasing

length, i.e. $N_1 < \cdots < N_Q$, so we can create a multiscale summary, by obtaining for each scale the corresponding storyboard. For simplicity, we also assume a step of one image between scales and $Q = N$, which provides the finest possible granularity to the multiscale summary. We compute the representative image of a cluster C, selected as the image with the lowest average distance to the rest of the images in the cluster. We find the index of the representative image as

$$\operatorname{rep}(C) = \underset{x, I_x \in C}{\arg\min} \sum_{I_y \in C, y \neq x} d(I_x, I_y) \tag{3}$$

where I_x and I_y are images in C, and $d(x, y)$ is their distance in the feature space.

The next step is to encode the ranked list. Given a set $\mathcal{H}$, the ranked list can be obtained using Algorithm 3. This algorithm is valid for any set of clusterings with increasing number of clusters, and any clustering algorithm can be used, as long as it can be tuned to generate different scales with different lengths. Obviously, the more scales the more computation effort.

Algorithm 3 Clustering-based list generation.

Input: set of clusterings $\mathcal{H}$, number of scales Q
Output: ranked list L

// Compute storyboards
for $q \leftarrow 1 : Q$ **do**
  $G^{(q)} \leftarrow \emptyset$ *// indices of the images in* $S^{(q)}$
  for $k \leftarrow 1 : \left|C_k^{(q)}\right|$ **do**
    Compute $g^* = \operatorname{rep}\left(C_k^{(q)}\right)$
    Set $G_k^{(q)} \leftarrow g^*$
  end for
end for
// Compute list
$L \leftarrow G^{(1)}$
for $q \leftarrow 2 : Q$ **do**
  Find $T = G^{(q)} \setminus G^{(q+1)}$ *// Deletions set*
  for all $g \in T$ **do**
    Append $(-1, g)$ to L
  end for
  Find $Y = G^{(q+1)} \setminus G^{(q)}$ *// Insertions set*
  for all $g \in Y$ **do**
    Append g to L
  end for
end for
return L

An interesting case is hierarchical clustering, which given the set of keyframes V naturally generates a hierarchy of nested clusterings

$$\mathcal{H} = \left\{C_1^{(1)}, \left\{C_1^{(2)}, C_2^{(2)}\right\}, \cdots, \left\{C_1^{(N)}, \cdots, C_N^{(N)}\right\}\right\}$$

where $C_i^{(q)}$ is the ith cluster at level q, assuming a step of one frame. This makes this clustering approach particularly suitable in our case, because it only requires one pass to obtain many scales with different lengths. Moreover, in the case of conventional agglomerative clustering, two consecutive clusterings with $q - 1$ and q clusters share

$q-2$ clusters, since only one cluster is split into two clusters. Thus, $q-2$ representative images are reused in the next scale. The remaining cluster $C_i^{(q-1)}$ is split into $C_j^{(q)}$ and $C_p^{(q)}$. Then there are three possible options: the representative of $C_i^{(q-1)}$ is also the representive of $C_j^{(q)}$, the representive of $C_p^{(q)}$, or none of them (i.e. a different keyframe). A consequence is that in a transition from $q-1$ to q, most of the images are preserved and only a maximum of one deletion would be required, which makes the ranked list compact and leads to scalable storyboards (not strictly scalable according to our previous definition). Following the previous observation, Algorithm 3 can be modified to be more efficient by avoiding most of the computation of representative images and finding difference sets, as shown in Algorithm 4. Besides, it can also be easily integrated in the clustering procedure. For the clustering implementation we chose the agglomerative clustering algorithm with single linkage[37]. Due to hierarchical clustering, the time complexity is still approximately $O\left(N^3\right)$, although Algorithm 4 is more efficient than Algorithm 3.

Algorithm 4 Hierarchical clustering-based list generation.

Input: hierarchy of clusterings $\mathcal{H}$, N
Output: ranked list L

Compute $g_1^{(1)} = \text{rep}\left(C_1^{(1)}\right)$
$L \leftarrow g_1^{(1)}$
for $q \leftarrow 2 : N$ **do** *// $C_i^{(q-1)}$ is split in $C_j^{(q)}$ and $C_p^{(q)}$*
    Compute $g_p^{(q)} = \text{rep}\left(C_p^{(q)}\right)$ and $g_j^{(q)} = \text{rep}\left(C_j^{(q)}\right)$
    if $g_i^{(q-1)} = g_p^{(q)}$ **then**
        Append $g_j^{(q)}$ to L *// Only $g_j^{(q)}$ is new*
    else if $g_i^{(q-1)} = g_j^{(q)}$ **then**
        Append $g_p^{(q)}$ to L *// Only $g_p^{(q)}$ is new*
    else
        Append $\left(-1, g_i^{(q-1)}, g_j^{(q)}, g_p^{(q)}\right)$ to L
    end if
end for
return L

5 Adaptation model and applications

5.1 Adaptation model

Multiscale storyboards are useful in situations in which the detail of the abstract depends on external constraints (e.g. display area) or the result of user interactions. Thus, navigation can be formulated as an adaptation problem, in which we want to maximize the amount of information presented in the storyboard, while satisfying certain constraints. We consider a model with two dimensions:

- *Spatial scale*, representing the different size of each keyframe when it is presented in the display area (e.g. image width in pixels or millimeters).

– *Summarization scale*, related to the amount of information presented. It can be measured using a convenient unit (e.g. number of images).

To include user interaction, each of the previous dimensions can be connected to a corresponding navigation action (see Fig. 3): *spatial zoom* and *semantic zoom*, respectively.

5.2 Adaptation to heterogeneous devices

As videos and images, summaries should be also adapted to the specific usage context. In particular the area available to display storyboards and an appropriate image size mainly depend on the screen size. In general, specific layouts and designs are also necessary to adapt the user interface to the variety of screen sizes and aspect ratios found in devices such as smartphones and tablets. Besides, most of these devices also allow portrait and landscape orientations. Scalable summaries provide a flexible way to adapt the summary to the specific requirements.

Spatial and summarization scales are related to the level of detail in the information presented to the user. The most detailed case would present all the keyframes with high resolution, but this is not a feasible solution as the display area is limited. Thus, the main external constraint is the *effective area* available to present the storyboard, which depends on the screen resolution and the user interface (note that, even for the same device, landscape and portrait user interfaces may have different effective areas). The effective area for presentation is fixed, so in practice this operation must also decrease (increase) the size of each image, trading off detail between the summarization scale and the spatial scale. Thus, given a rectangular canvas with effective area WxH, a size for elemental images wxh and the number of columns M_c, we can compute the number of rows required as

$$M_r = \left\lfloor M_c \frac{w}{h} \frac{H}{W} \right\rfloor = \left\lfloor M_c \frac{a}{A} \right\rfloor \tag{4}$$

where $a = \dfrac{w}{h}$ and $A = \dfrac{W}{H}$ are the respective aspect ratios. In practice a is constant, and the images are cropped accordingly (e.g. to the central region). The number of images in the summary is $M = M_c M_r$. Fig. 2a shows examples of adaptation to different devices and orientations.

5.3 Video navigation

While the characteristics of the devices impose some external constraints to the adaptation problem, the user can also change these constraints dynamically. A conventional *spatial zoom* would change the size of the images, acting over the spatial scale. Following the idea of semantic zooming in zoomable interfaces, we now define a *semantic zoom* operation over the summarization scale, which increases (decreases) the number of images in the storyboard, providing a more detailed (coarser) summary. Note that the summary still covers the whole sequence. Fig. 3 illustrates these two operations.

We can also include the *rotation* operation, which physically rotates the device and switches between portrait and landscape orientations. Fig. 2a shows an example of navigation trajectory in these coordinates using the previous operations, in which

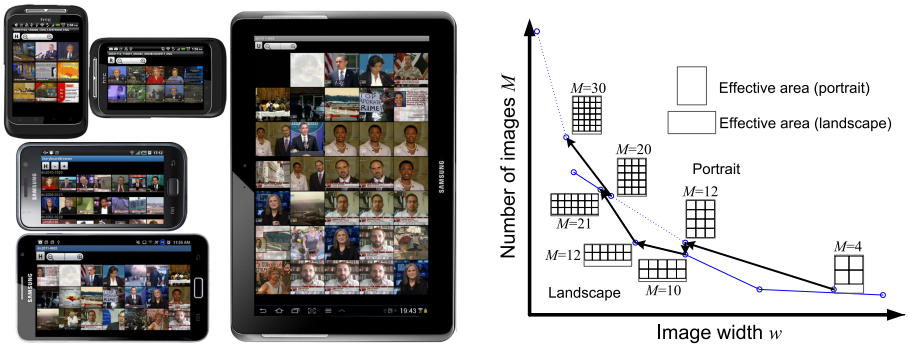

Fig. 2 Adaptation using scalable storyboards: (a) different devices and orientations, and (b) adaptation path in the spatial-summarization plane, via the navigation operations scaling and rotation. Feasible adaptation points for portrait and landscape orientations are shown.

Fig. 3 Zoom in storyboards: (a) spatial (spatial scale), and (b) semantic (summarization scale).

the user moves along the adaptation constraint (zoom) or jumps between portrait and landscape curves (rotation). In the latter, the closest point in the other curve, in terms of number of images and image width, is selected. Note that in this two-dimensional case, we are always assuming that the storyboard covers the whole video, and the solution of this problem is the number of images M.

A mayor concern for practical usability of scalable storyboards is that transitions from one scale to another scale should be as smooth as possible, with minimal change if possible, in order to avoid distracting effects caused by images disappearing in the new scale and other layout rearrangements (e.g. layout disturbance[15]). Fig. 4 illustrates this problem, comparing two different ways to create the storyboard, two resulting scales and the rearrangement of images resulting from the transition. In the first row the keyframes are those included based on the temporal order. We observe that it is relatively easy to follow the transition, as all the images coming from the previous scale are grouped together and stay at the beginning, and new images appear at the bottom right after them. In the case of uniform sampling (bottom row), it is much more difficult to follow them, as old images may reappear at arbitrary positions and new images can also appear at any position. Moreover, many images also dissapear during the transition. Thus, not only the user will not find those images, but also the number of new images is higher. These undesired factors increase user's cognitive load during interaction and impairs effective and comfortable interaction. Thus, the multiscale navigation structure and the user interface should take this effect into account.

Fig. 4 Example illustrating the transition from 3 columns to 4 columns storyboards using: a-c), temporal order, d-f), uniform sampling. c) and f) show how the images from the previous scale are relocated after the transition. While temporal order only covers the initial part of the content, the changes during the transition are relatively easy to follow. Uniform sampling covers the whole video, but the resulting changes are more difficult to track and even several images disappear in the new scale.

5.4 Other applications

In addition to generic video navigation and adaptation to mobile devices, scalable summaries can be useful in other applications. In particular they are very suitable when users need to find a specific piece of content in large collections of video data. Scalable summaries provide different degrees of detail which users can interactively adjust to find more detailed information in specific videos and segments. Examples of such tasks are the TRECVID known-item-search[34,35] or finding information in large video archives[13].

6 Evaluating multiscale storyboards

In this section we introduce the challenge of evaluating multiscale storyboards, by reviewing previous evaluation methods used in the literature to evaluate conventional (single scale) summaries, and analyzing the implications of these approaches when including multiple scales.

6.1 Properties of a good summary

There is certain agreement in that a summary should be compact (to be visualized quickly), have good coverage (including most of the relevant semantic information of the original video) and easy to visualize (the user must be comfortable visualizing the summary, which should be also free of spatial and temporal artifacts created by the summarization process). Thus we consider three main properties in a summary: compactness, coverage and smoothness. Note that other attributes have been also proposed, such as informativeness, interestingness, pleasantness, enjoyability or redundancy. We do not consider them as they are more or less related to the previous ones.

6.2 Evaluation with multiple scales

In general, there is no agreed protocol to evaluate video summaries. In the case of multiple scales an exhaustive evaluation would require each summary (for each method) is evaluated not only once but multiple times, which can have a dramatic impact on the cost of the evaluation (especially when users are involved). In the video summarization literature many methods have been used for evaluation[40]. We review the most common approaches and discuss their possible extension to multiple scales.

6.2.1 Result description

The proposed technique (and perhaps other baselines) is applied to a few sequences and the resulting storyboard is presented and described. The advantages of the proposed method are described over those examples. The evaluation is very subjective and often only reflects the authors' (biased) perspective. Although sometimes it can be useful to highlight some particular properties of the proposed technique and its behavior, provides little experimental evidence of why the proposed method is better.

This method can be easily extended to multiple scales. The main drawback is that it would require much more space in the paper to present the summaries, and describing and comparing different methods and scales may not be very easy. In any case, it would still have the same inconveniences as in the single scale case.

6.2.2 Objective metrics

Some kind of metric that automatically compares the original video and the summary providing a score reflecting fidelity or estimating user satisfaction would be much more convenient. Some methods use low-level visual features and compare the selected images and the original keyframes or video using some objective metric. Often, the particular evaluation metric is often the same that the method tries to optimize, and there is no guarantee that that metric correlates well with the actual subjective utility. Summarization is a cognitive process involving high level understanding to select semantically important content and discard irrelevant. Thus, it is not easy to obtain an appropriate evaluation metric based only on low-level features.

In certain domains, the objective of the summary is to present compactly the highlights or important events (e.g. sports). In this case, a relatively objective ground truth can be obtained by labeling representative segments, and summaries can be evaluated using information retrieval metrics such as precision and recall. This approach can

be easily extended to multiple scales, computing these metrics at different scales and comparing methods based on the resulting curves.

An alternative when determining the important events is to let users manually create summaries (often several users are necessary, as the summary will depend on the particular user) and then use them as ground truth to compare with the summaries generated by the system. Unfortunately, the extension to multiple scales would require the users to create as many summaries as scales, which increases the cost significantly.

6.2.3 User studies

Perhaps the most accepted approach to evaluate summaries is a direct evaluation using user studies. These studies provide more valuable insight, as the purpose of summaries is to eventually be used by users. Thus, the opinion of users is the most valuable and realistic information in an evaluation. However, these evaluations are not always available because of a more difficult setting, a much higher cost (e.g. collecting volunteers) and often it is not easy to formulate properly the questions to be asked (sometimes different users have different interpretations). In contrast to objective measures based on some fixed ground truth, user studies cannot be replicated to compare with new methods, often requiring to carry out a new user study.

A typical evaluation session would require users to visualize a number of summaries of the same video (as many as methods to compare). Then they are requested to rate the summary according to certain criteria (which are usually related with properties such as coverage, redundancy, pleasantness, etc). Then this process is repeated for each video in the evaluation set, and then for each user.

A proper design also requires taking into account other external factors that may have impact over the scores, especially when evaluating many summaries leads to long evaluation sessions. Factors such attention and fatigue often cause scores vary accross the evaluation session itself, thus the score of a particular summary may have some bias related with the order of visualization. While the scores at the beginning of the evaluations may be more carefully chosen, users may unintentionally change their scoring as the session becomes long and the evaluation task tedious. The experimental design should take into account these factors by splitting long sessions into shorter ones, or designing the order to balance some biases.

While, as we just discussed, evaluating conventional summaries is already complex, in the case of multiple scales this complexity increases significantly. For some applications one or very few scales are enough, so exhaustive evaluation may still be practical. For others, a much finer granularity is desirable. In this case, the number of combinations is very high and makes evaluation very time consuming and impractical, unless the trials are limited to few scales.

As we described before, it is different to evaluate a summary of an unfamiliar video than evaluating a summary of a video we already know. Exhaustive evaluation requires that for each video, several summaries are visualized and rated (e.g. one per summarization method, one per scale). The more summaries of a given video have been visualized and rated, the more information is known, so the scores may be affected by the visualization order and introduce bias. This problem is even more severe in dynamic summaries, such as scalable video skims. There are several ways to try to alleviate this problem. We can design the experiment in such a way that each user visualizes only one scale and one summary per video (different users evaluate different scale/version). Then the design of the experiment must take care of that all the combinations are covered

and that the bias is canceled or at least equally distributed (e.g. using Latin squares in the design). However, this requires many more users and also it is more sensitive to problems during the evaluation session. Another option is to let the user visualize all the videos prior to the evaluation of the summaries. In this way the user is already familiar with the content. But when the evaluation includes several 30 minutes or 1 hour videos this is impractical. In sum, to cope with these factors (i.e. bias, fatigue, etc), a proper evaluation requires a very complex experimental setting and long evaluation sessions, which is often not possible in practice.

6.2.4 Task oriented evaluation

Some authors evaluate the utility of a summary for certain specific task (e.g. seeking a specific scene), rather than the summary itself. In this case, users are often required to complete a set of tasks, but they do not provide a subjective score but other indirect measurable properties are used instead, such as clicks, log specific actions or time required to complete the task.

In this case the user study is evaluating the usability of the whole system rather than the utility of the summaries themselves. In many tasks, an intuitive and effective design is often more critical than the summary itself. So in this case it is hard to decouple user interface and summaries, and thus obtain an unbiased comparison of summarization methods.

6.3 Proposed evaluation metrics

We are targeting a particularly challenging scenario, i.e. a large number of scales with a step of one keyframe between consecutive ones. In this scenario, exhaustive user studies are impractical, so we would like to design some objective metrics that help us at least to get some useful insight about the behavior of the different methods.

For convenience, we first define several concepts we will use in the evaluation measures:

- *Informative image (II)*: image conveying new information, and different from the information conveyed by other images in the summary. For instance, a blank image is a non-informative image. A near duplicate keyframe of a previous one is also considered non-informative.
- *Semantic group (SG)*: group of keyframes conveying the same information. For instance, all the images from the anchorperson segment belong to the same semantic group.

6.3.1 Information

In general, it is difficult to determine which images are the *best* images for the storyboard, as it depends on each particular user. In practical browsing situations exploring dozens of unknown videos, users often do not care whether the images shown are the most representative, they just expect images showing diverse information to get a quick idea of what the content is about and decide. However, including very similar images (e.g. same anchorperson in Fig. 5) or non-informative images is easily perceived as redundant and a waste of display area. For these reasons, we evaluate how informative

the summaries are rather than how representative they are, with the following two measures:

- *Ratio of informative images*: measures the ability of the algorithm to provide informative images, or in other words, to avoid redundant and non-informative images. The *ratio of informative images* of a summary at a certain scale is computed as

$$RII\,(M) = \min\left(\frac{\#II}{M}, 1\right) \tag{5}$$

  where M is the number of images in the summary at that scale and $\#II$ is the number of informative images.
- *Semantic coverage*: measures how well the summary covers the set of keyframes from the semantic point of view. It is similar to the informativeness measure but weighted by the size of the semantic group with related images. The *semantic coverage* of a summary at a certain scale is computed as

$$SC\,(M) = \frac{1}{N^{'}} \sum_{\forall G_i, I_m \in G_i, I_m \in S^{(M)}} |G_i| \tag{6}$$

  where M is the number of images in the summary at that scale $S^{(M)}$, $|G_i|$ is the size of the semantic group G_i, measured in number of keyframes, and N' is the total number of keyframes excluding the non-informative ones.

6.3.2 Transitions between scales

Considering now the application of scalable summaries to navigation (see Section 5.3), we also proposed a task-oriented metric that can help to estimate which methods are more suitable in this scenario. As we discussed previously, an important aspect related with the usability of scalable storyboards in navigation is that the transition between different scales should be smooth, trying to preserve most information between scales, so the new information (i.e. new keyframes) can be tracked more easily. In order to evaluate this aspect we propose the *ratio of preserved images* (RPI) that measures the amount of common information shared by two scales with M and M' images

$$RPI\left(M, M'\right) = \frac{\left|S^{(M)} \bigcap S^{(M')}\right|}{M} \tag{7}$$

As this ratio requires defining both an initial scale and a final scale, different settings are possible. A fine grained incremental setting may evaluate this ratio when the number of images in two consecutive scales differ in one. Another possibility is to evaluate this ratio within a specific user interface and device. We consider these two settings in our experiments.

7 Experiments

We performed a series of experiments to compare the different methods described previously, focusing on the particular characteristics of multiscale storyboards and in the context of smartphones. As in this paper we focus on the framework rather than on design aspects of the user interface, we leave more specific task-oriented and usability studies to further work. We include some visual examples to illustrate the behavior of the different methods, and also include both objective and subjective metrics.

7.1 Experimental Setup

For the experiments we focused on news video content, using two different set of videos:

- *dn*: ten news program videos available at the Internet Archive. A typical video of this collection contains anchorpersons, reports and relatively long interviews, which make the set of keyframes relatively redundant. This collection is very suitable to illustrate the differences between the different methods studied in the paper.
- *tv*: ten videos from the TrecVid 2005 dataset[38], including five CNN and five MSNBC news videos. These videos are more dynamic than those in the previous collection.

We evaluated four methods: *temporal*, *uniform*, *hierarchical* and *ranking*, corresponding to the methods described in Section 4. These methods were implemented in the Android platform, using native code for feature extraction (color histograms in HSV space and Euclidean distance, which we found fast and satisfactory enough for the videos in the datasets). For the objective evaluation we also included an additional method *kmeans*, which uses K-means to obtain the scalable description from a set of clusterings. The effective area to display storyboards was 480x650 (portrait) and 800x330 (landscape). The number of images in V was set to $N = 60$ (roughly one and two keyframes per minute for the *dn* and *tv* dataset, respectively). This provides us with a reasonable amount of keyframes and fine-grained scales (i.e. 60 scales) and still not too many to provide insightful visual comparisons (those in Section 7.2). Apart from these reasons, sampling a keyframe every minute (or 30 seconds) is also reasonable in a practical application with these datasets.

7.2 Comparison of the methods

A visual comparison of the methods illustrates easily the difference between them. We took some snapshots of the interface, with different methods and different scales. Results for the video browsing interface are shown in Fig. 5. We can observe how different methods select images in very different ways. The *temporal* method usually selects the introduction and often an initial blank image (videos in *dn* dataset sometimes start with blank screen). The content is very redundant even for few images. In contrast, *uniform* distributes the images along the whole sequence, which is often satisfactory in the case of short summaries. For videos with long interviews and recurrent scenes, and for longer summaries it also tends to select redundant images. Obviously it is not possible to avoid these effects without any content analysis. Both *hierarchical* and *ranking* provide more variety in the selection of images at different scales. The difference is more significant at intermediate scales. The most detailed scale shows the whole keyframe set V (in average for both datasets, only around 50% of the images in V are informative).

7.3 Objective Evaluation

We also made a quantitative comparison using the proposed metrics to evaluate how the different methods are able to keep useful information and how graceful the transitions between scales are.

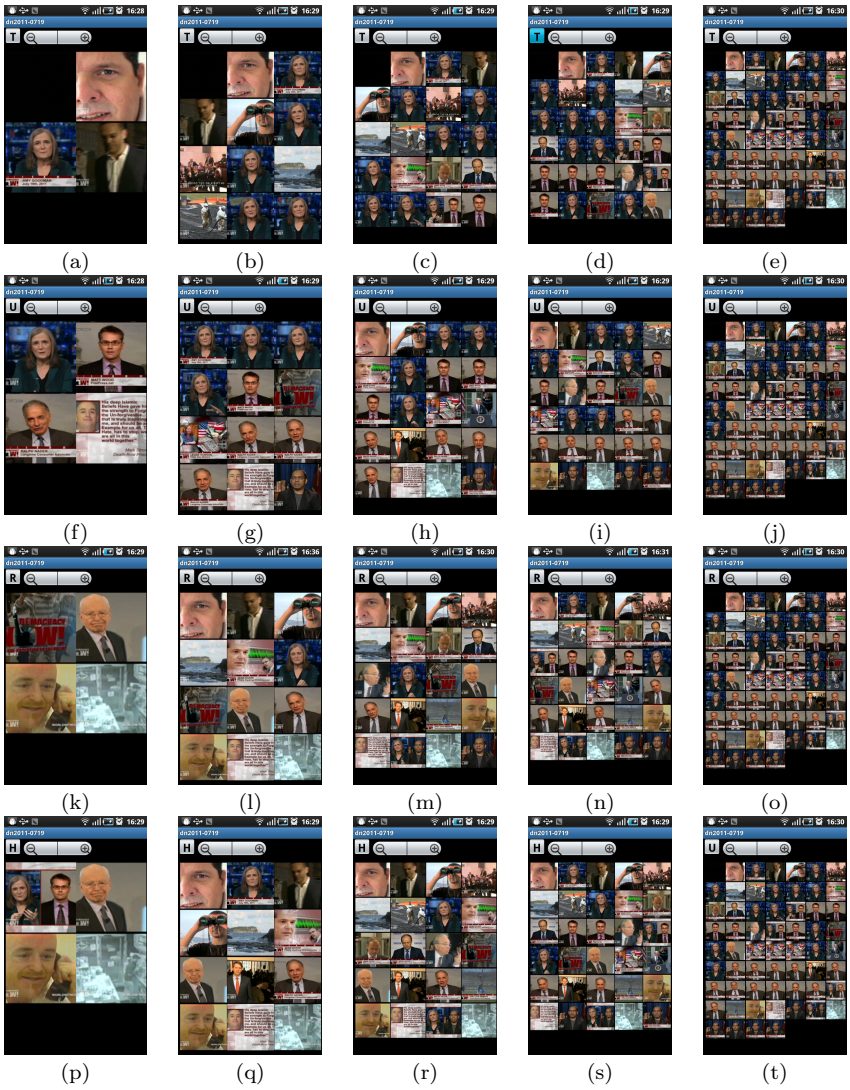

Fig. 5 Results for single item (2, 3, 4, 5 and 7 columns), *dn* dataset: a-e), temporal order, f-j), uniform sampling, k-o), scalable storyboard (ranking), p-t), scalable storyboard (hierarchical). The 7 columns summaries show all the keyframes.

7.3.1 Information

The ground truth was obtained by manually annotating the set of keyframes. As news videos are relatively structured, we consider different semantic groups (e.g. anchorperson, certain interviewee, synthetic tables, maps). All the images in the same semantic group were given the same label, and a special label for non-informative images, such as blank images, which do not belong to any semantic group.

The average ratio of informative images in each dataset is shown in Fig. 6. As the original sequences are already redundant, there is a limit in how many informative

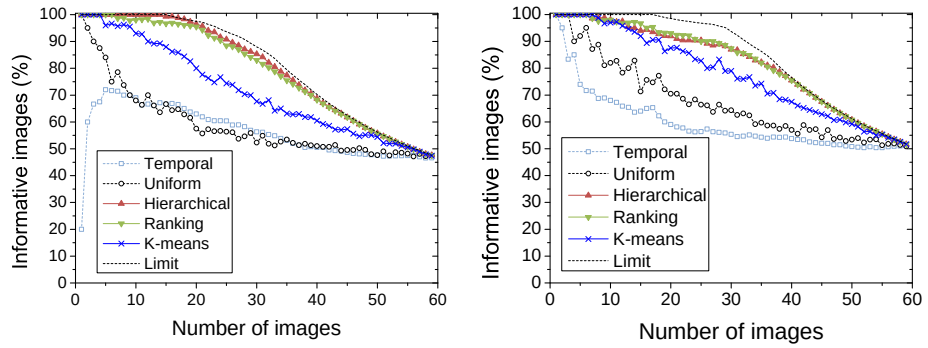

Fig. 6 Informativeness: a) *dn* dataset, b) *tv* dataset.

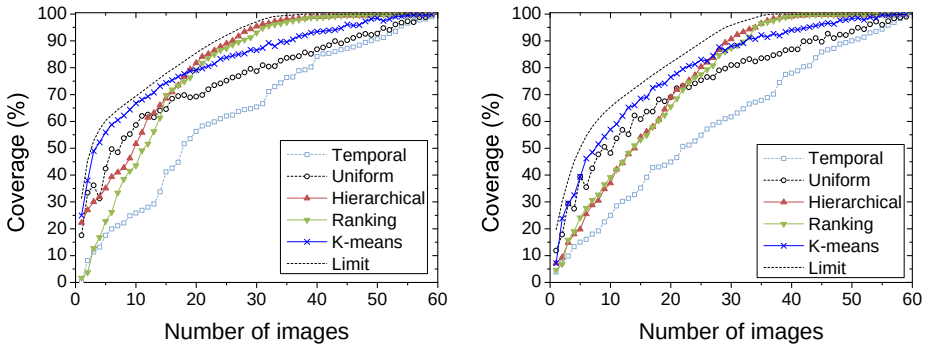

Fig. 7 Semantic coverage: a) *dn* dataset, b) *tv* dataset.

images can be selected at a certain level. Using the ground truth we can compute this limit, thus giving a measure of the redundancy of the dataset. In Fig. 6a we observe that *temporal* selects fewer informative images at low scales for the *dn* dataset, due to blank images in the first seconds of the videos. We did not remove those blank images to keep the method fully content-blind. In the case of the *tv* dataset (see Fig. 6b) there are no blank images, so the performance is still reasonable at low scales, but it quickly degrades. For *uniform*, the problem of initial blank frames is not present, as the images are sampled at different points. The results improve slightly, avoiding repetitions at initial intervals. However, as the number of images increases, redundant images are also selected, and it tends to have a similar performance to the temporal order method. For a large number of images is inevitable to select redundant images. In the case of content-based methods, the redundancy is better exploited, being the results much closer to the limit. The performance of *hierarchical* is slightly better than that of *ranking* in the *dn* dataset, and in the case of the *tv* dataset there is no significative difference. If instead of using hierarchical clustering we use K-means clustering (*kmeans* method) the results are not good according to this measure.

However, in terms of semantic coverage (see Fig. 7), the method that performs better is *K-means*, being close to the limit at low scales. With a larger number of images, *hierarchical* and *ranking* have better performance. Clustering methods emphasize the coverage aspect, useful to represent more information with fewer images, with K-means having better performance. In the case of *dn* dataset, *hierarchical* also outperforms *ranking*, as iterative ranking only focuses on selecting dissimilar images. Surprisingly, the performance of *uniform* at low scales is better that both *scalable* and *hierarhical*, even being a content-blind method.

7.3.2 Transition between scales

We evaluate the suitability of the summarization methods for a navigation. Considering a step of one keyframe between scales, Fig. 8a and b show the RPI in both datasets, with similar results. In this aspect both *temporal* and *ranking* are optimal, as they explicitly use the previous set and include one new image each time. In the case of hierarchical, the result is also very close, as two consecutive clusterings only differ in one cluster. Thus, for a step of one keyframe at most only two keyframes may be different. The least appropriate method in this aspect is *uniform*, which does not impose any restriction and the selection rule implicitly tends to redistribute the selected keyframes with little overlap with the previous sampling instants. In contrast to hierarchical clustering, the resulting clusterings with K-means are independent, so *kmeans* method does not preserve as many images between consecutive scales.

In our interface, users vary the number of columns, rather than images, so the transitions include larger steps in practice. Fig. 8c and d show the same measure evaluated for the particular test device. The results and conclusions are similar.

7.4 Subjective Evaluation

We also conducted a user study in order to evaluate the different methods and also whether the proposed objective metrics are related to the subjective feedback provided by users. A total of 11 volunteers participated in the experiment. The perceived utility of different combinations of methods and number of columns was assessed using a Likert scale (1: disagree; 3: neutral; 5: agree)[22] with the statement "*Considering the number of images available at a certain scale (X columns), the summary is an adequate representation of the content.*". This statement explicitly emphasizes that the summaries must be evaluated taking the limitation in length into account. In preliminary tests we observed that if this aspect is not explicitly mentioned, some users evaluate them in a relative way, and some users in an absolute way. With the latter interpretation, users tend to systematically give low scores to short summaries and high scores to long summaries. A total of 24 storyboards were evaluated for each video (4 methods and 6 scales, from 1 to 6 columns).

The results are shown in Fig. 9a and b. In general, users tend to have a clear preference for content-based scalable methods, especially at intermediate scales, for which the difference is more significant. We can observe similar trends to those in the RPI measure (compare with Fig. 6). *temporal* generates storyboards with low utility, especially for low scales due to blank images in the *dn* dataset. Without the problem of blank images (*tv* dataset), the perceived utility still degrades very quickly. The most useful comparison is between the *uniform* and content-based methods. In this case, *uniform*

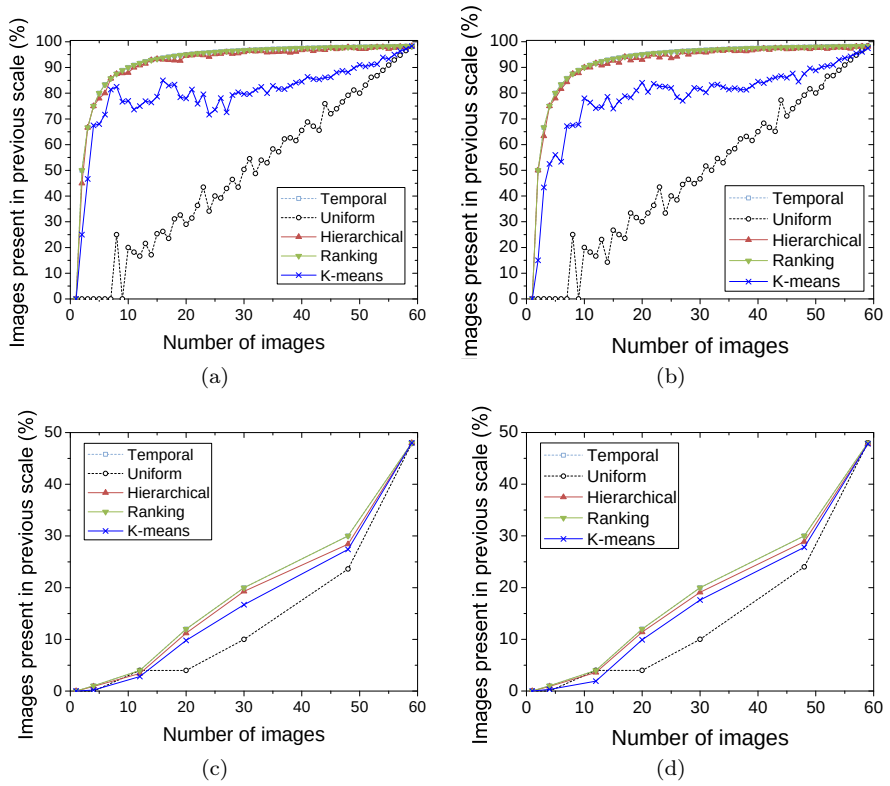

Fig. 8 Number of images present at previous scale: a) *dn* dataset, step 1 keyframe, b) *tv* dataset, step 1 keyframe, c) *dn* dataset, step 1 column, and d) *tv* step 1 column.

rarely selects blank frames (see Fig. 5f-j), so they do not have practical influence on the result. This method can be satisfactory at low scales, but the utility degrades as the number of images increases, resulting in more redundancy. The perceived utility is significantly higher with content-based method, although, as expected, it degrades slightly at higher scales when the remaining redundant images are included. Comparing the content-based methods, the difference is small, with *hierarchical* having slightly better results than *ranking*.

Fig. 9c and d shows the average utility (six scales) for each video. We see that content-based methods are very consistent and with satisfactory results in all the videos. The results for *uniform* and *temporal* are more irregular, varying largely from from video to video, but always with lower scores than the content-based methods.

8 Discussion and conclusions

In this paper we focus on a few little explored aspects of video summarization, such as scalable summaries, their application to flexible navigation and adaptation in handheld devices, and the challenge of evaluating the quality and usability of summaries with multiple scales. In particular we consider multiscale storyboards that can be dynam-

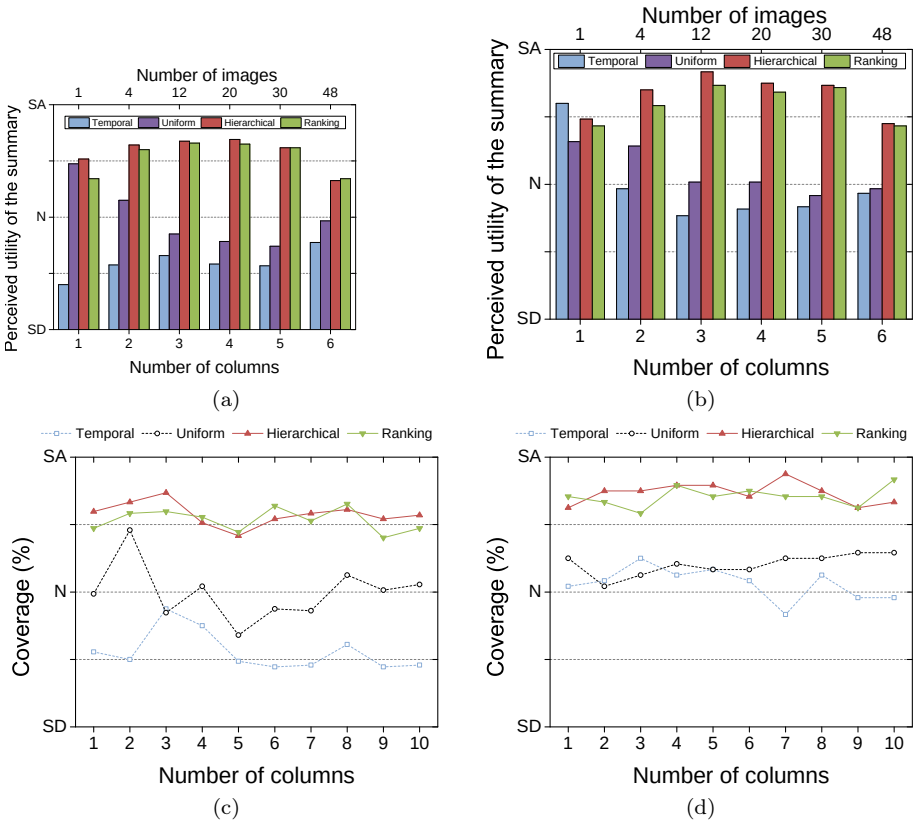

Fig. 9 User study (left column for *dn* and right for *tv*): a-b) average per number of columns.

ically adapted to a required length with a very fine granularity. All the information to recover the summary with a given length is encoded in a compact descriptor, i.e. ranked list, which is precomputed and thus it does not require any costly process during adaptation. Many applications could benefit from this flexibility. In particular we study the application to enable enhanced summary adaptation and navigation in handheld devices.

Although in this paper we use relatively simple features and methods, more complex analysis methods including higher levels clues (e.g. faces) and temporal structuring (e.g. shots, scenes) can be adopted to obtain better ranked lists. In addition, domain-specific analysis can be also included. For instance, dialog scenes can be identified and exploited in TV series and comedy movies. Although complex analysis may not be suitable for handheld devices, this can be done remotely in the server and the result stored as a ranked list. In any case, the user interface can benefit from the flexibility of scalable summaries and the proposed navigation and adaptation model.

One important difference in scalable summaries is that the result is not a single summary anymore, but a collection of summaries with increasing length, and more sophisticated summarization algorithms should proceed accordingly. The role of the length is very important. While low level features and redundancy removal may be

satisfactory for medium size summaries, higher level semantic and structural analysis (e.g. scenes, dialogs) is often essential for very short summaries. A smart scalable summarization algorithm should be able to vary the summarization criteria across this wide range of scales. Nevertheless, we showed that we can obtain relatively satisfactory storyboards in practice for news videos, even using simple methods and low level features.

Evaluation of scalable summaries still needs further exploration. Exhaustive evaluations with user studies are impractical, especially for fine grained summaries, and alternative evaluation approaches are necessary. We proposed some simple objective metrics that can help to model the quality and usability of summaries, which can be used to compare different summarization methods. These metrics approximately follow similar trends as those found in our user study. We also emphasize interscale properties, which are specific to the multiscale nature of these representations, and must be evaluated if the application requires transitions between scales. In general, format-specific aspects should be also considered, such as audio and temporal properties in scalable video skims[18,8] or the implications of variable layout in scalable comic-like summaries[15].

A critical aspect to fully exploit the potential of scalable storyboards is the design of effective user interfaces and intuitive interaction models. Scalable summaries are useful tools that can make navigation flexible and easily adaptable. However they are useless if the interface is not intuitve or if the interaction is uncomfortable. So far, several algorithms to create scalable summaries have been proposed in the literature, but there is little research studying and evaluating them in an application context. In the case of storyboards, an appropriate design should exploit the advantages of scalability for flexible navigation (e.g. semantic zoom), while minimizing the cognitive burden during transitions between scales, due to potential changes in sizes and locations of images.

Acknowledgements This work was supported in part by the National Basic Research Program of China (973 Program): 2012CB316400, in part by the National Natural Science Foundation of China: 61322212 and 61350110237, in part by the National Hi-Tech Development Program (863 Program) of China: 2014AA015202, and in part by the Chinese Academy of Sciences Fellowships for Young International Scientists: 2011Y1GB05. This work was also funded by Lenovo Outstanding Young Scientists Program (LOYS).

References

1. Adami, N., Signoroni, A., Leonardi, R.: State-of-the-art and trends in scalable video compression with wavelet-based approaches. IEEE Transactions on Circuits and Systems for Video Technology **17**(9), 1238–1255 (2007)
2. Ahmad, I., Wei, X., Sun, Y., Zhang, Y.Q.: Video transcoding: an overview of various techniques and research issues. IEEE Transactions on Multimedia **7**(5), 793–804 (2005)
3. Albanese, M., Fayzullin, M., Picariello, A., Subrahmanian, V.: The priority curve algorithm for video summarization. Information Systems **31**(7), 679–695 (2006)
4. Benini, S., Bianchetti, A., Leonardi, R., Migliorati, P.: Extraction of significant video summaries by dendrogram analysis. In: Proc. IEEE International Conference on Image Processing, pp. 133–136 (2006)
5. Benini, S., Migliorati, P., Leonardi, R.: Statistical skimming of feature films. International Journal of Digital Multimedia Broadcasting **2010**, 11 (2010)
6. Bescos, J., Martinez, J.M., Herranz, L., Tiburzi, F.: Content-driven adaptation of on-line video. Signal Processing: Image Communication **22**, 651–668 (2007)

7. Chang, S.F., Vetro, A.: Video adaptation: concepts, technologies, and open issues. Proceedings of the IEEE **93**(1), 148–158 (2005)
8. Cong, Y., Yuan, J., Luo, J.: Towards scalable summarization of consumer videos via sparse dictionary selection. IEEE Transactions on Multimedia **14**(1), 66–75 (2012)
9. Dong, P., Xia, Y., Wang, S., Zhuo, L., Feng, D.: An iteratively reweighting algorithm for dynamic video summarization. Multimedia Tools and Applications pp. 1–25 (2014). DOI 10.1007/s11042-014-2126-8
10. Dumont, E., Merialdo, B.: Rushes video summarization and evaluation. Multimedia Tools and Applications **48**, 51–68 (2010)
11. Friedland, G., Gottlieb, L., Janin, A.: Narrative theme navigation for sitcoms supported by fan-generated scripts. Multimedia Tools and Applications **63**(2), 387–406 (2013). DOI 10.1007/s11042-011-0877-z
12. Gong, Y., Liu, X.: Video summarization using singular value decomposition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 174–180 (2000)
13. Haesen, M., Meskens, J., Luyten, K., Coninx, K., Becker, J., Tuytelaars, T., Poulisse, G.J., Pham, P., Moens, M.F.: Finding a needle in a haystack: an interactive video archive explorer for professional video searchers. Multimedia Tools and Applications **63**(2), 331–356 (2013)
14. Hanjalic, A., Zhang, H.: An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. IEEE Transactions on Circuits and Systems for Video Technology **9**(8), 1280–1289 (1999)
15. Herranz, L., Calic, J., Martínez, J.M., Mrak, M.: Scalable comic-like video summaries and layout disturbance. IEEE Transactions on Multimedia **14**(4), 1290–1297 (2012)
16. Herranz, L., Martinez, J.M.: Generation of scalable summaries based on iterative GOP ranking. In: Proc. IEEE International Conference on Image Processing, pp. 2544–2547 (2008)
17. Herranz, L., Martínez, J.M.: An integrated approach to summarization and adaptation using H.264/MPEG-4 SVC. Signal Processing: Image Communication **24**(6), 499–509 (2009)
18. Herranz, L., Martínez, J.M.: A framework for scalable summarization of video. IEEE Transactions on Circuits and Systems for Video Technology **20**(9), 1265–1270 (2010)
19. Hürst, W., Darzentas, D.: History: a hierarchical storyboard interface design for video browsing on mobile devices. In: Proc. International Conference on Mobile and Ubiquitous Multimedia, pp. 17:1–17:4 (2012)
20. Irie, G., Satou, T., Kojima, A., Yamasaki, T., Aizawa, K.: Automatic trailer generation. In: Proceedings of the International Conference on Multimedia, MM '10, pp. 839–842. ACM, New York, NY, USA (2010)
21. Li, Y., Merialdo, B.: Vert: automatic evaluation of video summaries. In: Proceedings of the international conference on Multimedia, pp. 851–854 (2010)
22. Likert, R.: A technique for the measurement of attitudes. Archives of Psychology **22**(140), 1–55 (1932)
23. Liu, H., Liu, Y., Sun, F.: Video key-frame extraction for smart phones. Multimedia Tools and Applications pp. 1–19 (2014). DOI 10.1007/s11042-014-2390-7
24. Marchionini, G., Wildemuth, B.M., Geisler, G.: The Open Video digital library: A Möbius strip of research and practice. Journal of the American Society for Information Science and Technology **57**(12), 1629–1643 (2006)
25. Mohanta, P.P., Saha, S.K., Chanda, B.: Generation of size constrained video storyboard using spanning tree. In: Proceedings of the First International Conference on Internet Multimedia Computing and Service, pp. 179–182 (2009)
26. Money, A.G., Agius, H.: Video summarisation: A conceptual framework and survey of the state of the art. Journal of Visual Communication and Image Representation **19**(2), 121–143 (2008)
27. Mukherjee, D., Said, A., Liu, S.: A framework for fully format-independent adaptation of scalable bit streams. IEEE Transactions on Circuits and Systems for Video Technology **15**(10), 1280–1290 (2005)
28. Mundur, P., Rao, Y., Yesha, Y.: Keyframe-based video summarization using delaunay clustering. International Journal of Digital Libraries **6**(2), 219–232 (2006)
29. Ohm, J.R.: Advances in scalable video coding. Proceedings of the IEEE **93**(1), 42–56 (2005)
30. Over, P., Smeaton, A.F., Awad, G.: The TRECVid 2008 BBC rushes summarization evaluation. In: Proc. 2nd ACM TRECVid Video Summarization Workshop, pp. 1–20. ACM (2008)

31. Over, P., Smeaton, A.F., Kelly, P.: The TRECVid 2007 bbc rushes summarization evaluation pilot. In: TRECVid
32. Santini, S.: Who needs video summarization anyway? In: Proc. Int. Conf. Semantic Computing ICSC 2007, pp. 177–184 (2007)
33. Scherp, A., Mezaris, V.: Survey on modeling and indexing events in multimedia. Multimedia Tools and Applications **70**(1), 7–23 (2014). DOI 10.1007/s11042-013-1427-7
34. Schoeffmann, K.: A user-centric media retrieval competition: The video browser showdown 2012-2014. MultiMedia, IEEE **21**(4), 8–13 (2014). DOI 10.1109/MMUL.2014.56
35. Schoeffmann, K., Ahlstrom, D., Hudelist, M.: 3-d interfaces to improve the performance of visual known-item search. Multimedia, IEEE Transactions on **16**(7), 1942–1951 (2014). DOI 10.1109/TMM.2014.2333666
36. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Transactions on Circuits and Systems for Video Technology **17**(9), 1103–1120 (2007)
37. Sibson, R.: Slink: an optimally efficient algorithm for the single-link cluster method. The Computer Journal **16**(1), 30–34 (1973)
38. Smeaton, A.F., Over, P., Kraaij, W.: Evaluation campaigns and TRECVid. In: Proc. of the ACM international workshop on Multimedia information retrieval, pp. 321–330. ACM (2006)
39. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(12), 1349–1380 (2000)
40. Truong, B.T., Venkatesh, S.: Video abstraction: A systematic review and classification. ACM Trans. on Multimedia Computing, Communications and Applications **3**(1), 3 (2007)
41. Valdes, V., Martinez, J.M.: Automatic evaluation of video summaries. ACM Trans. Multimedia Comput. Commun. Appl. **8**(3), 25:1–25:21 (2012)
42. Vetro, A.: MPEG-21 digital item adaptation: Enabling universal multimedia access. IEEE Multimedia **11**(1), 84–87 (2004)
43. Xin, J., Lin, C.W., Sun, M.T.: Digital video transcoding. Proceedings of the IEEE **93**(1), 84–97 (2005)
44. Yang, Y., Ma, Z., Xu, Z., Yan, S., Hauptmann, A.G.: How related exemplars help complex event detection in web videos? In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013, pp. 2104–2111 (2013)
45. Yuan, Z., Lu, T., Wu, D., Huang, Y., Yu, H.: Video summarization with semantic concept preservation. In: Proc. International Conference on Mobile and Ubiquitous Multimedia, pp. 109–112. ACM (2011)
46. Zhang, L., Gao, Y., Hong, C., Feng, Y., Zhu, J., Cai, D.: Feature correlation hypergraph: Exploiting high-order potentials for multimodal recognition. Cybernetics, IEEE Transactions on **44**(8), 1408–1419 (2014). DOI 10.1109/TCYB.2013.2285219
47. Zhao, S., Yao, H., Sun, X.: Video classification and recommendation based on affective analysis of viewers. Neurocomputing **119**, 101 – 110 (2013)
48. Zhao, S., Yao, H., Sun, X., Jiang, X., Xu, P.: Flexible presentation of videos based on affective content analysis. In: S. Li, A. El Saddik, M. Wang, T. Mei, N. Sebe, S. Yan, R. Hong, C. Gurrin (eds.) Proc. International Conference on Multimedia Modeling, *Lecture Notes in Computer Science*, vol. 7732, pp. 368–379 (2013)
49. Zhu, X., Fan, J., Elmagarmid, A.K., Wu, X.: Hierarchical video content description and summarization using unified semantic and visual similarity. Multimedia Systems **9**(1), 31–53 (2003)
50. Zhu, X.Q., Wu, X.D., Fan, J.P., Elmagarmid, A.K., Aref, W.F.: Exploring video content structure for hierarchical summarization. Multimedia Systems **10**(2), 98–115 (2004)
51. Zhuang, Y., Rui, Y., Huang, T., Mehrotra, S.: Adaptive key frame extraction using unsupervised clustering. In: Proceedings of International Conference on Image Processing, pp. 866–870 (1998)