Video Abstraction: A Systematic Review and Classification

BA TU TRUONG and SVETHA VENKATESH
Curtin University of Technology

The demand for various multimedia applications is rapidly increasing due to the recent advance in the computing and network infrastructure, together with the widespread use of digital video technology. Among the key elements for the success of these applications is how to effectively and efficiently manage and store a huge amount of audio visual information, while at the same time providing user-friendly access to the stored data. This has fueled a quickly evolving research area known as *video abstraction*. As the name implies, video abstraction is a mechanism for generating a short summary of a video, which can either be a sequence of stationary images (*keyframes*) or moving images (*video skims*). In terms of browsing and navigation, a good video abstract will enable the user to gain maximum information about the target video sequence in a specified time constraint or sufficient information in the minimum time. Over past years, various ideas and techniques have been proposed towards the effective abstraction of video contents. The purpose of this article is to provide a systematic classification of these works. We identify and detail, for each approach, the underlying components and how they are addressed in specific works.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey—; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information System—*Video (e.g., tape, disk, DVI)*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*

General Terms: Algorithms, Management

Additional Key Words and Phrases: Video summarization, video abstraction, keyframe, video skimming, survey

ACM Reference Format:

Truong, B. T. and Venkatesh, S. 2007. Video abstraction: A systematic review and classification. ACM Trans. Multimedia Comput. Commun. Appl. 3, 1, Article 3 (Feb. 2007), 37 pages. DOI = 10.1145/1198302.1198305 http://doi.acm.org/10.1145/1198302.1198305.

1. INTRODUCTION

The revolution in digital video witnessed in recent years has been driven by the rapid development in various areas of computer infrastructure such as increased processing power, bigger and less expensive capacity of storage devices, and faster networks. This revolution has brought many new applications and, as a consequence, research into new technologies that aim at improving the effectiveness and efficiency of video acquisition, archiving, cataloging and indexing, as well as increasing the usability of stored videos.

A video sequence normally contains a large number of frames. In order to ensure that humans do not perceive any discontinuity in the video stream, a frame rate of at least 25fps is required, that is, 7,500 images for one hour of video content. This sheer volume of video data is a barrier to many practical

Authors' address: B. T. Truong and S. Venkatesh, Department of Computer Science, Curtin University of Technology, Perth, WA, Australia; email: {truongbt,svetha}@cs.curtin.edu.au.

applications, and therefore, there is a strong demand for a mechanism that allows the user to gain certain perspectives of a video document without watching/addressing the video in its entirety. This mechanism is termed *video abstracting*.

Abstraction techniques are mainly designed to facilitate browsing of a video database. They complement the automatic video retrieval approach (i.e., searching), especially when content-based indexing and retrieval of video sequences has only seen limited success. There are many semantic concepts that are very difficult or even impossible to extract automatically, and in other cases, precise queries cannot be easily formed. The major limitation of video abstraction browsing is that it can only be effective if the number of target video sequences is relatively small. Browsing through a large number of video abstracts to find a desired sequence can be very time-consuming and tiresome to the user. Fortunately, the two approaches can be combined in the same manner as an Internet search engine in helping the user find relevant information. The video search engine serves as a filter of content of the video database according to the user query, which is either textual or content-based, and the abstracts of potentially relevant video sequences are returned. The user then can browse through these abstracts to locate the desired videos.

In addition to facilitating browsing on a large video collection, video abstracts also help the user to navigate and interact with one single video sequence in a nonlinear manner analogous to the video editing storyboard. They enable quick access to a semantically relevant position in a video sequence. This is particularly useful in video editing and authoring applications. Video abstracts can also be used as an end product to be shared, digested, and enjoyed by the user. Retaining only the essential information of a video sequence improves storage, bandwidth, and viewing time. Typical applications are sport highlight programs on television. On the computational side, a concise representation of a video sequence significantly reduces the computational overhead of many video content analysis and retrieval tasks, for example, object/face detection and segmentation.

Recognizing the importance of video abstraction within the broader field of multimedia content management, there has been, in recent years, a significant body of work that seeks automated and objective solutions to the generation of video abstracts. The developed techniques target video data from various domains, such as movies, documentaries, sports, news, home videos, e-learning, etc., and address various viewpoints and assumptions on what constitutes a good or optimal video summary. Promising results have been reported, especially in the generation of sports highlights. However, video abstraction is still largely in the research phase; practical applications are still limited in both complexity of method and scale of deployment. For example, video search services such as Yahoo and Alta Vista use a single keyframe to represent the video, while Google provides a context-sensitive keyframe list of the video. Static storyboards are helping the browsing process in Open-Video Archive,[1] while David Gibson and Thomas [2002] are working towards using keyframes with the existing text-based query engine to speed-up the process of finding relevant video tapes for worldwide researchers in the library of wildlife footage at the BBC's Natural History unit. Virage[2] offers a successful system that provides customized highlights to baseball games in major league baseball. Otsuka et al. [2005] recently developed a prototype personal video recorder that allows the detection, retrieval, and playback of sports highlights, while Wan et al. [2005] are experimenting with delivering sports highlights to mobile phone users on the G3 network. A search on the US patent and trademark office website[3] shows that some video abstraction techniques have also been patented, for example, Mauldin et al. [1997] from CMU and Leinhart and Yeo [2003] from Intel Corporation.

[1]www.open-video.org
[2]www.virage.com
[3]www.uspto.gov

The purpose of this work is to provide a comprehensive overview of existing solutions to the problem of video abstraction. We focus more on problem formulation, as well as the features and techniques developed, and less on the discussion regarding the strengths, limitations, and performance of a specific work. This is because a meaningful characterization of these aspects is almost impossible, since experimental setups and summarization viewpoints are often considerably different. Nevertheless, we provide general comments on specific aspects of the video abstraction process based on our experience in multimedia content management. Although overviews of some existing approaches have been reported elsewhere [Li et al. 2001; Kang 2002], to our knowledge, this is the most comprehensive and systematic review of the field to-date. Unlike other reviews, we focus strongly on identifying critical aspects of video abstraction, ranging from problem formulation to result evaluation, analyzing and classifying how these are addressed in various works.

The layout of this article is as follows. In the next section, we describe the differences and associations between two main types of video abstracts: *keyframes* and *video skims*. Techniques for generating these two types of video abstracts will be discussed in Sections 3 and 4, respectively. In each section, we provide a uniform formulation of the problem, and describe important aspects of a video abstraction solution, such as the data domain, dominant features, and underlying computational mechanisms, thus, facilitating a classification of existing techniques according to the manner in which these aspects are addressed. In Section 5, we describe existing methods for evaluating the performance of video abstraction techniques and outline some practical recommendations towards a consistent evaluation framework for the field. Section 6 concludes this article.

2. VIDEO ABSTRACTS

In this work, we consider two basic forms of video abstracts:

—*Keyframes*. These are also called *representative frames*, *$\mathcal{R}$-frames*, *still-image abstracts* or *static storyboard*, and a set consists of a collection of salient images extracted from the underlying video source. Hence, the keyframe set $\mathcal{R}$ is defined as follows:

$$\mathcal{R} = \mathcal{A}_{Keyframe}(\mathbf{V}) = \{f_{r_1}, f_{r_2}, \ldots, f_{r_\mathbf{k}}\}, \tag{1}$$

where $\mathcal{A}_{keyframe}$ denotes the keyframe extraction procedure.

—*Video skim*. This is also called a *moving-image abstract*, *moving storyboard*, or *summary sequence*. This type of abstract consists of a collection of video segments (and corresponding audio) extracted from the original video. These segments are joined by either a cut or a gradual effect (e.g., fade, dissolve, wipe). It is itself a video clip, but of significantly shorter duration. One popular kind of video skim in practice is the movie trailer. The video skim $\mathcal{K}$ is defined as follows:

$$\mathcal{K} = \mathcal{A}_{Skim}(\mathbf{V}) = \mathbf{E}_{i_1} \odot \mathbf{E}_{i_2} \odot \ldots \odot \mathbf{E}_{i_\mathbf{k}}, \tag{2}$$

where $\mathcal{A}_{Sim}$ denotes the skim generation procedure. $\mathbf{E}_i \sqsubset \mathbf{V}$ is the ith excerpt to be included in the skim, and $\odot$ is the excerpt assembly and integration operation (e.g., cut, fade, dissolve, wipe). Note that the preceding representation is applicable to the video stream only, whereas the audio stream of a video skim can be resequenced, edited, and does not necessarily synchronize with the original video. This issue will be addressed in detail later in this article.

One advantage of a video skim over a keyframe set is the ability to include audio and motion elements that potentially enhance both the expressiveness and information of the abstract. In addition, it is often more entertaining and interesting to watch a skim than a slide show of keyframes. On the other hand, since they are not restricted by any timing or synchronization issues, once keyframes are extracted, there are further possibilities of organizing them for browsing and navigation purposes, rather than

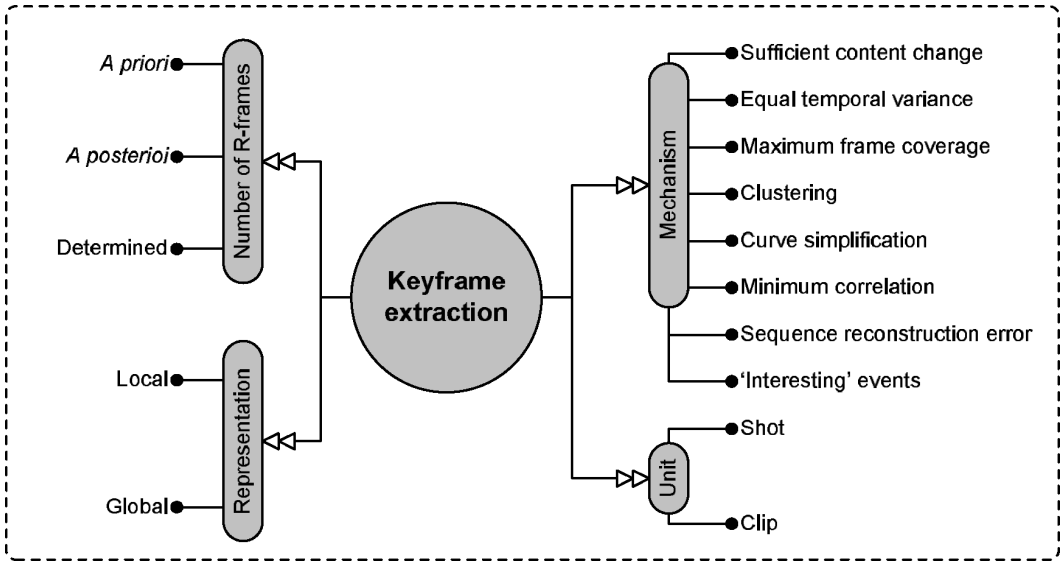

Fig. 1. Attributes of keyframe extraction techniques.

the strict sequential display of video skims, as demonstrated in Tonomura et al. [1993], Yeung and Leo [1997], Uchiashi et al. [1999], and Girgensohn et al. [2001], and the transformation of keyframes temporal order to spatial order allows the user to grasp the video content more quickly, whilst sacrificing screen space. Keyframes are also useful in reducing the computational complexity for various video analysis and retrieval applications.

Note that although video skims and keyframes are often generated differently, these two forms of video abstract can be transformed from one to the other. Video skims can be created from keyframes by joining fixed-size segments, subshots, or the whole shots that enclose them, as employed in Hanjalic and Zhang [1999], Wu et al. [2000], and Lee and Haye [2004]. On the other hand, the keyframe set can be created from the video skim by uniform sampling or selecting one frame from each skim excerpt.

3. KEYFRAMES

The simplest method for keyframe generation is uniform sampling. Although simple and computationally efficient, sampling-based methods may produce no keyframes for a short, yet semantically important, segment, whilst producing too many keyframes with identical content to represent a long static segment, thus failing to effectively represent the actual video content. Our review of keyframe extraction techniques will focus only on techniques that take into account the underlying dynamics, to different degrees and from varying viewpoints, of the video sequence. During the review process, we have uncovered fundamental aspects of the current approaches in keyframe extraction, as depicted in Figure 1. These aspects are: the size of the keyframe set, the base unit, representation scope, the underlying computational mechanisms, and visualization method for extracted keyframes. For each aspect, we provide a categorization of how they are addressed in existing works and describe their pros and cons.

3.1 The Size of the Keyframe Set

There are different options for determining the number of keyframes in an automatic keyframe extraction process, and they strongly shape the underlying formulation of the optimal keyframe set. The size of the keyframe set can be fixed as a known *priori*, left as an unknown *posteriori*, or determined

internally within the abstraction process. Most of the techniques only offer one option for the size of keyframe set. However, if the algorithm produces a number of keyframes progressively [Divakaran et al. 2002; Liu et al. 2004], two options are available: It can stop when the number of keyframes reaches *a priori* value or when certain criteria are satisfied (i.e., *a posteriori*). Li et al. [2004] also highlight that for applications where efficient utilization of network bandwidth and storage is crucial, the size of keyframe set should be measured by the total number of bits required to code the summary with a given coding strategy.

3.1.1 *A Priori.* The number of keyframes is decided beforehand and given as a constraint to the extraction algorithm. It can be assigned as a specific number or ratio over the length of the video that may vary according to user knowledge of the video content. Also called *rate constraint keyframe extraction*, this approach is suitable and often required in mobile device systems where available resources are limited. For these systems, the number of keyframes are distributed differently, depending on the transmission bandwidth, storage capacity, or display size of the receiving terminal. A special yet common case is when one keyframe is selected per shot, which is often the first frame, middle frame, or that frame closest to the average content of the shot (also see Section 3.2). This technique has a disadvantage in that it does not ensure that all important segments in a video contain at least one keyframe.

Ideally, the keyframe extraction problem with *a priori* size $\mathbf{k}$ can be formulated as an optimization problem of finding the frame set $\mathcal{R} = \{f_{i_1}, f_{i_2}, \ldots, f_{i_\mathbf{k}}\}$ that is differs least from the video sequence with respect to a certain summarization perspective:

$$\{r_1, r_2, \ldots, r_\mathbf{k}\} = \arg\min_{r_i}\{\mathcal{D}(\mathcal{R}, \mathbf{V}, \rho) \mid 1 \leq r_i \leq \mathbf{n}\}, \tag{3}$$

where n is the number of frames in the original video sequence, ρ is the summarization perspective that the user is interested in, and $\mathcal{D}$ is a dissimilarity measure.

Most current keyframe extraction techniques have ρ as "visual coverage," which aims to cover as much visual content with as few frames as possible; however, ρ can also be the number of objects, number of faces, etc. It can also represent a viewpoint on what constitutes the optimal keyframe set, and shape the underlying computational solution (see Section 3.4).

3.1.2 *A Posteriori.* In this approach, we do not know the number of extracted keyframes untill the process finishes. The number of keyframes is often determined by the level of visual change, itself. More keyframes are needed to represent a video sequence with a lot of action and movements than required for a static one. However, an excessively large number of keyframes may be produced for highly dynamic scenes, causing inconvenience during interactive tasks. The formulation of the keyframe extraction problem with no specified size requires a dissimilarity tolerance ε, also called the *fidelity* level. This can generally be formulated as

$$\{r_1, r_2, \ldots, r_\mathbf{k}\} = \arg\min_{r_i}\{\mathbf{k} | \mathcal{D}(\mathcal{R}, \mathbf{V}, \rho) < \varepsilon, 1 \leq r_i \leq \mathbf{n}\}. \tag{4}$$

Technically, with a significant increase in computational cost, this approach can be converted into the first approach (number of keyframes is *a priori*) by iteratively reducing the fidelity level until the number of keyframes produced approximates a predefined value, and vice versa.

3.1.3 *Determined.* This is in essence the *a posteriori* approach; however, we need to acknowledge the approaches that attempt to determine the appropriate number of keyframes before the full extraction is executed. For example, with cluster-based methods, cluster validation can be performed to determine the number of clusters before or within the actual clustering process [Hammoud and Mohr 2000; Ferman and Tekalp 2003; Yu et al. 2004]. Hanjalic et al. [1998], [Liu et al. 2003], and Fauvet et al. [2004], on

Table I. Summary of Notation

Symbol	Description
$\mathbf{V}$	A video clip or segment (e.g., shot/scene) on which the summary is independently extracted
f_i	The ith frame of the video sequence
$\mathbf{n}$	The number of frames in $\mathbf{V}$, $\mathbf{n} = \|\mathbf{V}\|$
$\mathcal{R}$	The set of all keyframes of $\mathbf{V}$
$\mathbf{k}$	The number of keyframes or video excerpts (of a skim)
r_i	The index of i-th keyframe of $\mathbf{V}$
$[b_i, b_{i+1})$	The boundaries of the segment being represented by keyframe f_{r_i}
$\mathcal{K}$	The video skim extracted from $\mathbf{V}$
$\odot$	Excerpt assembly or integration operation that joins individual excerpts to create a video skim
$\mathbf{E}_i$	The i-th excerpt of a video skim
ρ	The summarization perspective
$\mathcal{D}(.)$	A dissimilarity/difference function, symmetric
$\mathcal{C}(f_i, f_j)$	A content change function, not necessarily symmetric

the other hand, consider the number of keyframes allocated for each shot as being proportional to its accumulative content change.

3.2 Unit

An important issue is to determine what base temporal *unit* the keyframe represents. In the literature, keyframes can be extracted to represent individual shots as an immediate step or directly represent the entire clip. The first method is often classified as *shot-based* and requires shot indices. The straightforward shot-based technique is to select the first, last, and/or middle frame of a shot as its keyframes. However, most methods, as described in Section 3.4, are adaptive to the visual dynamics of the shot. Shot-based approaches are intuitive in the way that shots are commonly understood as the primitive unit of meaning in video. They are also aided by the fact that shot boundary detection has been concluded to be a solved problem by the NIST TRECVID benchmark,[4] since a highlevel of performance has been consistently obtained over time and on a large dataset [Kraaij et al. 2004]. The second approach, on the other hand, may proceed without knowledge of shot boundaries and is termed *clip-based* (e.g., Girgensohn and Boreczky [1999], Yu et al. [2004]). Here, the term clip may also refer to structural elements of higher order than the shot, such as, scenes and story units.

In shot-based methods, the shots are processed independently of each other; therefore, each shot is treated as a separate clip and the $\mathbf{V}$ notation (see Table I) is applied throughout this section. Furthermore, unless some postprocessing is applied, a shot-based method may produce keyframes that are similar to each other due to the presence of similar shots in the video sequence, whereas clip-based methods generally produce a more compact set of keyframes. For video browsing systems, using one or more frames to represent a shot does not scale up for long videos, since the presentation may take up the whole screen space and looking through a large number of images is time-consuming, tedious, and ineffective. On the other hand, shot-based methods are generally more efficient, as only a small chunk of frames is processed at a time.

3.3 Representation Scope of Individual Keyframes

This aspect of the keyframe extraction algorithm relates to whether an extracted keyframe is representative of its local neighbourhood segment or represents noncontiguous segments of a video clip. The second approach tends to produce a smaller keyframe set, while the first preserves the temporal visual progression, which may be important for understanding the action and events in a summary. For

[4]www-nlpir.nist.gov/projects/trecvid/.

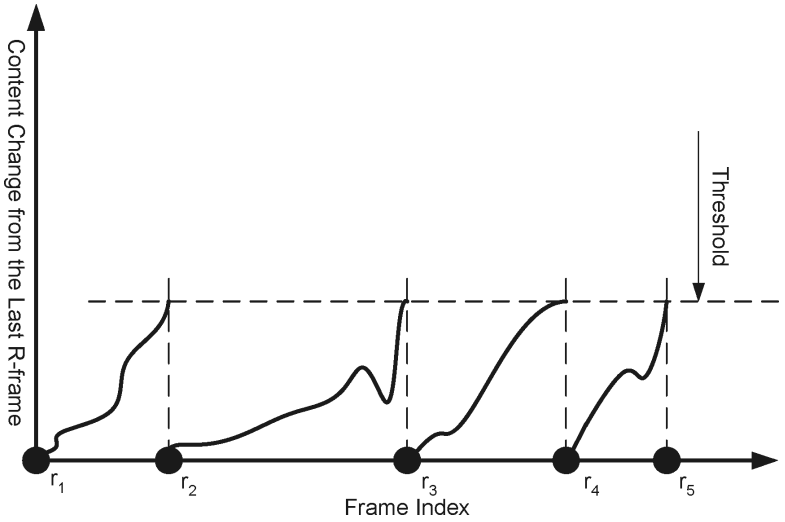

Fig. 2. Illustration of the sufficient content change method.

example, the keyframes extracted in Xiong et al. [1997] have a local representation scope, each keyframe representing the content of the video sequence from its temporal position until the next keyframe. On the other hand, clustering-based methods such as Zhuang et al. [1998] and Yu et al. [2004] tend to produce keyframes with global representation scope, each representing the visual content of all frames in its cluster that may spread across the video shot/sequence.

3.4 Underlying Computational Mechanisms

We categorize the underlying computational mechanisms for keyframe extraction into eight somewhat overlapping classes, as depicted in Figure 1.

3.4.1 *Sufficient Content Change.* This method proceeds sequentially and only requires knowledge about the video sequence up until the current temporal position, selecting a frame as the keyframe only if its visual content significantly differs from previously extracted keyframes (see Figure 2). In the other words, in its basic form, the *sufficient content change* method selects the next keyframe $f_{r_{i+1}}$ based on the most recently selected keyframe f_{r_i} as follows:

$$r_{i+1} = \arg\min_{t}\{\mathcal{C}(f_t, f_{r_i}) > \varepsilon, i < t \leq \mathbf{n}\}, \tag{5}$$

with the first frame of the base unit, for example, the shot, often selected as the first keyframe, namely, $r_1 = 1$.

As for the content change function, a variety of metrics have been proposed in the literature, the most popular being the histogram difference as used in Yeung and Leo [1995], Zhang et al. [1997], and Kang et al. [1999]. Alternative metrics include the accumulated energy function computed from image-block displacements across two successive frames [Zhang et al. 2003] and changes in the number or geometric properties of extracted video objects [Kim and Hwang 2002]. The latter is more applicable when video objects can be extracted effectively and efficiently, as in the case of surveillance videos [Kim and Hwang 2002]. The selection of appropriate metric is crucial not only to this, but to almost all other approaches described in this article.

One problem associated with the sufficient content change method as presented so far is that the keyframe does not represent any portion of the video sequence preceding it and hence, its representation

coverage is not maximized. Xiong et al. [1997] propose an extension to the method called *seek and spread*, which does not require the reference frame (to locate the next keyframe) and current keyframe to be the same. Let b_i denote the index of the current reference frame ($b_1 = 1$). Then, r_i and b_i are updated as

$$r_i = \arg\min_t \{\mathcal{C}(f_t, f_{b_i}) > \varepsilon, t > b_i\} \tag{6}$$

$$b_{i+1} = \arg\min_t \{\mathcal{C}(f_t, f_{r_i}) > \varepsilon, t > r_i\}. \tag{7}$$

In this approach, the keyframe f_{r_i} represents the segment $[b_i, b_{i+1} - 1]$ of the video sequence.

The sufficient content change method can be a method of choice for many real-time and/or online applications due to its simplicity and intuitiveness. It has the desired property that video shots/sequences with different lengths and activity levels are represented with a varying number of keyframes and the positions of keyframes reflect their dynamic progression. However, while sequential processing ensures the efficient computational extraction of keyframes, as only $\mathbf{n}$ content change operations $\mathcal{C}(.)$ need be computed, it causes the produced keyframe set to have an unwanted asymmetric property. In addition, this method is essentially shot-based and extracted keyframes have local representative scope, and therefore suffer from the scalability issue for longer video sequences. In Rasheed and Shah [2003], a more concise set of keyframes is obtained by generating a new keyframe if it is sufficiently different not only to the last keyframe, but to all existing ones.

3.4.2 *Equal Temporal Variance.* This is somewhat a variant of the sufficient content change method, which requires that the number of keyframes is specified *a priori*, and assumes that a good keyframe set should have individual keyframes represent video segments $(b_i, b_{i+1} - 1)$ of equal temporal variance. In this approach, the determination of $\{b_i\}$ and $\{r_i\}$ is performed separately in two steps. Theoretically, segment boundaries $\{b_i\}$ are selected so that

$$\mathcal{V}(b_1, b_2) = \mathcal{V}(b_2, b_3) = \cdots = \mathcal{V}(b_{\mathbf{k}}, b_{\mathbf{k}+1}), \tag{8}$$

where $\mathcal{V}(b_i, b_{i+1})$ denotes the temporal variance of the segment $(b_i, b_{i+1} - 1)$, which has f_{r_i} as its representative frame. Since the exact equality of temporal variance is often not achived, Sun and Kankanhalli [2000] reformulate it as an optimization problem.

$$\{b_1, b_2, \ldots, b_{\mathbf{k}+1}\} = \arg\min_{b_i} \sum_{i=1}^{\mathbf{k}} \sum_{j=1}^{\mathbf{k}} |\mathcal{V}(b_i, b_{i+1}) - \mathcal{V}(b_j, b_{j+1})| \tag{9}$$

The temporal variance function $\mathcal{V}$ can be approximated by the cumulative content change across consecutive frames [Lee and Kim 2002; Divakaran et al. 2002; Fauvet et al. 2004] and Eq. (8) can be solved directly from the accumulative content curve. Furthermore, $\mathcal{V}$ can also be approximated by the difference between the first and last frames of the segment, and the optimization problem in Eq. (9) can be approximately solved by recursively deleting a set of consecutive frames with the smallest content change until the desired number of keyframes is obtained [Sun and Kankanhalli 2000].

Once the boudaries $\{b_i\}$ are determined, Divakaran et al. [2002] and Fauvet et al. [2004] simply select the first and middle frame of every segment as the keyframe, respectively. On the other hand, in Lee and Kim [2002], the frame f_{r_i} is selected from each segment $[b_i, b_{i+1} - 1]$ by minimizing the total difference between f_{r_i} to all frames in its representation scope.

In general, the equal temporal variance method is more computationally expensive than the sufficient content change method. However, the produced keyframe set is globally "optimal" and independent of any processing order.

3.4.3 *Maximum Frame Coverage.* Proposed by Chang et al. [1999] and also referred to as the fidelity-based approach, this method attempts to use the representation coverage (i.e., the list of frames that a

given frame can represent) of individual frames in the sequence as the starting point. Let $\mathbf{C}_i(\varepsilon)$ denote the set of frames in $\mathbf{V}$ that can use f_i as their representative with respect to a tolerance or fidelity value ε. Under this framework, the optimal set of keyframes from $\mathbf{V}$ with no rate constraint can be extracted as

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = \arg\min_{r_i}\{\mathbf{k} | \mathbf{C}_{r_1}(\varepsilon) \cup \mathbf{C}_{r_2}(\varepsilon) \cup \cdots \cup \mathbf{C}_{r_{\mathbf{k}}}(\varepsilon) = \mathbf{V}\}. \tag{10}$$

When a given number of keyframes is specified as a constraint, the problem can be interpreted as either finding the minimum fidelity value ε such that all frames can be covered by at least one keyframe

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = \arg\min_{r_i}\{\varepsilon | \mathbf{C}_{r_1}(\varepsilon) \cup \mathbf{C}_{r_2}(\varepsilon) \cup \cdots \cup \mathbf{C}_{r_{\mathbf{k}}}(\varepsilon) = \mathbf{V}\} \tag{11}$$

or finding the set of frames that can represent as many frames as possible.

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = \arg\max_{r_i}\{|\mathbf{C}_{r_1}(\varepsilon) \cup \mathbf{C}_{r_2}(\varepsilon) \cup \cdots \cup \mathbf{C}_{r_{\mathbf{k}}}(\varepsilon)|\} \tag{12}$$

Two crucial components in this approach are the determination of coverage for a frame and the optimization procedure. In Chang et al. [1999], the coverage $\mathbf{C}_i(\varepsilon)$ of frame f_i consists of all frames in $\mathbf{V}$ that are visually similar to f_i (including itself). Chang et al. [1999] aim at finding an approximate solution by employing a greedy method which extracts the frame with maximum coverage at each step, removing every frame in its coverage from the video sequence and updating the coverage table. This iterates until there is no frame left in the video sequence. A similar greedy procedure can be applied to approximately solve Eq. (12), that is, when the number of keyframes is set as *a priori*.

The maximum frame coverage principle also forms the basis of the solution proposed in Yahiaoui et al. [2001]. However, in their work, the coverage of a frame is the number of fixed-size excerpts (not individual frames) which contain at least one frame similar to this frame. The dynamic programming procedure is used to select a prespecified number of frames as the keyframe set. Nevertheless, the advantage of using fixed-size excerpts over individual frames is unclear and not justified in this work. Rong et al. [2004] and recently, Cooper and Foote [2005], on the other hand, assume the analogy between keyframe extraction and popular keyword extraction in text information retrieval via the term frequency-inverse document frequency method (TF-IDF). Therefore, for a frame to be considered the keyframe of a shot/scene, it needs to have good frame coverage for this shot/scene, but low frame coverage for the entire video sequence.

The advantage of the maximum frame coverage formulation is that it does not require the representation coverage of a frame to be a contiguous segment as in the sufficient content change and equal temporal variance methods, and therefore, can generate a more concise keyframe set. However, the dissimilarity needs to be computed on all frame pairs, making it computationally more expensive, and thus unsuitable for real-time and/or online applications. It also requires a true visual similarity metric to fully realize its optimality.

3.4.4 *Clustering.* This approach treats video frames as points in the feature space (e.g., color histogram) and works on the assumption that the representative points of clusters formed in this space can be used as keyframes for the entire video sequence. It does not rely on any explicit modeling in the form of Eqs. (3) and (4), while making use of generic techniques developed for data clustering. Clustering can be performed as clip-based or shot-based, and often involves the four following steps. Existing methods are distinguished based on how these steps are implemented:

—*Preprocessing the data.* Preprocessing the input data is aimed at improving the effectiveness and efficiency of the clustering process. In David Gibson and Thomas [2002] and Yu et al. [2004], each

video frame is transformed into the eigenspace via principal component analysis (PCA) and kernel PCA, respectively. The purpose of this transformation is to perform dimensionality reduction so that high-dimensional image space can be represented by a much lower dimension space, whilst retaining the significant variations of the original data. Xiong et al. [1997], on the other hand, extract a set of potential keyframes via the sufficient content change method and use them as the clustering input. Similarly, Girgensohn and Boreczky [1999] use only a predefined number of frames that are largely different to at least one of the adjacent frames as the input to the clustering algorithm. Their argument is that two adjacent frames that are not different to each other will likely end up in the same cluster.

—*Clustering the data.* Standard clustering techniques or their domain-dependent variants now can be applied to the data to identify potential clusters. Xiong et al. [1997] and Zhuang et al. [1998] employ a sequential clustering technique that assigns the current frame to an existing cluster if their similarity (using the L_1 measure) is maximum and exceeds a threshold, and create a new cluster otherwise. Girgensohn and Boreczky [1999] use the hierarchical complete link method, whilst Yu et al. [2004] employ the fuzzy c-mean clustering technique. Gaussian mixture models (GMMs) are used in David Gibson and Thomas [2002], in which the number of components (i.e., the number of keyframes) that best clusters the data is computed via an iterative process based on a maximum likelihood threshold.

—*Filtering clusters.* Since clustering outputs may be noisy and/or the cluster itself not sufficiently significant, keyframes are not assigned to all clusters. Zhuang et al. [1998] only consider clusters whose sizes are greater than the average size of clusters. Girgensohn and Boreczky [1999] suggest that a cluster is assigned the keyframe if it contains at least one uninterrupted nine-second sequence of frames so as to avoid video artifacts. In Uchiashi et al. [1999], cluster filtering is performed via an importance score computed from the size of the cluster and its commonness.

—*Extracting the representative points of each cluster.* The most common and intuitive solution is to select that point closest to the cluster centroid as the representative point of the cluster, which eventually forms the keyframe set of the video sequence [Yu et al. 2004]. Similarly, for each selected segment in Uchiashi et al. [1999], the frame closest to the segment center is selected as the keyframe. Likewise, cluster centroids are the centres of each component in the GMM approach [David Gibson and Thomas 2002].

It appears that clustering is a popular method for keyframe extraction due to its common use in data analysis. However, since semantic meaningful clusters in video data often exhibit large intraclass and low interclass visual variance, successful extraction of these clusters is very difficult. Existing works in keyframe extraction often fail to adequately evaluate the results of the clustering process. In addition, the clustering-based approach is generally not suitable when we want to preserve the temporal progression of the video sequence from extracted keyframes.

3.4.5 *Minimum Correlation Among Keyframes.* Techniques in this class often deal with the rate constrained keyframe extraction problem and work on the basis that the keyframe set should have minimal correlation among its elements (sometimes only the correlation among successive elements is considered). It aims to select frames that are dissimilar to each other. The optimal extraction of keyframes via the minimal correlation criterion can be formulated as

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = \arg\min_{r_i} \{\mathcal{C}orr(f_{r_1}, f_{r_2}, \ldots, f_{r_{\mathbf{k}}})\}, \tag{13}$$

where $\mathcal{C}orr(.)$ is a correlation measure.

Doulamis et al. [1998] consider the contribution of correlations among all individual frame pairs to the correlation value of the entire keyframe set, which is formulated as

$$\mathcal{C}orr(f_{r_1}, f_{r_2}, \ldots, f_{r_{\mathbf{k}}}) = \left(\sum_{i=1}^{\mathbf{k}-1} \sum_{j=i+1}^{\mathbf{k}} \mathcal{C}orr(f_{r_i}, f_{r_j})^2 \right)^{1/2}, \tag{14}$$

where $\mathcal{C}orr(f_i, f_j)$ is the correlation coefficient of two feature vectors (f_i, f_j). Different algorithms can be used to find a near-optimal solution, such as logarithmic search [Doulamis et al. 1998], scholastic search [Doulamis et al. 1999], and the genetic algorithm [Doulamis et al. 2000].

On the other hand, if only the correlations of successive elements are taken into account, the formulation in Eq. (13) can be rewritten as

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = \arg\min_{r_i} \left\{ \sum_{i=1}^{\mathbf{k}-1} \mathcal{C}orr(f_{r_i}, f_{r_{i+1}}) \right\}, \tag{15}$$

which relates it to the sufficient content change and equal temporal variance methods. However, the extraction of keyframes based on this formulation does not aim at dividing the sequence into uniform temporal dynamic segments, but rather endeavors to maximize the total dynamic difference. Liu and Kender [2002b] note the existence of the optimal substructure within this formulation. In other words, given $\mathcal{R}$ the optimal set of keyframes for $\mathbf{n}$ frames, $\mathcal{R} - f_{r_{\mathbf{k}}}$ has the maximum correlation among all subsequences of size $\mathbf{k} - 1$ of $\mathbf{V}$ that ends in $f_{r_{\mathbf{k}-1}}$. Thus, a dynamic programming process with $(O(n^3))$ computation time can be used to extract the optimal solution. Liu and Kender [2002b] further propose an approximate solution based on a greedy algorithm to further reduce the computation time to $(O(n \log(n))$. This algorithm starts with all $\mathbf{n}$ frames as keyframes at level $\mathbf{n}$. As it goes from level t to $t - 1$, the frame with minimal correlation to its neighboring keyframes is dropped.

While the minimum frame correlation criterion ensures a low level of redundancy in the keyframe set, it suffers greatly from outliers. Porter et al. [2003] attempt to overcome this problem by treating the keyframe set as a path in a weighted directed graph, where every frame in the video sequence is represented by a node and the edge represents the correlation between two frames, which is computed from the accumulative motion parameter vectors. The existence of the edge is subject to a set of conditions that enforce the temporal ordering of the keyframe set and certain overlapping between two successive keyframes, essential for predicting the content unfolding in between. This method requires the first and last frames to be in the keyframe set, and the optimal keyframe set is therefore the shortest path from the first vertex/frame to the last, as in Eq. (15), which can be found via the A^* search strategy.

3.4.6 *Sequence Reconstruction Error (SRE)* . This approach is proposed in Liu and Kender [2002a], Lee and Kim [2003], and Liu et al. [2004], and is based on a metric called the SRE score (also known as the *shot reconstruction degree*). This measures the capability of the keyframe set for reconstructing the original video sequence/shot. It is useful when the number of keyframes is set as *a priori* and the temporal progression is crucial to the designated application. Assume that we have a frame interpolation function $\mathcal{I}(t, \mathcal{R})$ that calculates the full image or some characterizing features at time t in the video sequence from the keyframe set. The SRE score $\mathcal{E}(\mathbf{V}, \mathcal{R})$ of the keyframe set is defined as

$$\mathcal{E}(\mathbf{V}, \mathcal{R}) = \sum_{i=1}^{\mathbf{n}} \mathcal{D}(f_i, \mathcal{I}(i, \mathcal{R})), \tag{16}$$

where $\mathcal{D}(.)$ computes the difference between two image frames.

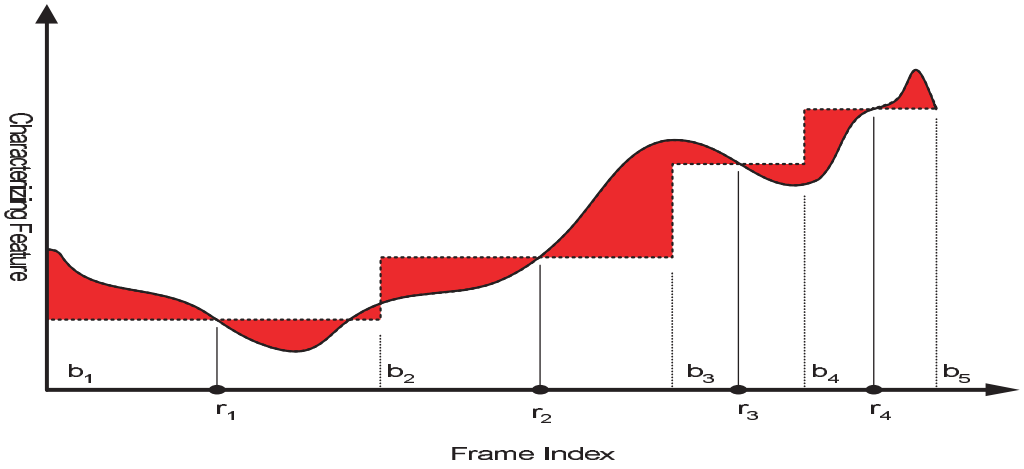

Fig. 3. Illustration of a sequence reconstruction error.

Given a rate constraint $\mathbf{k}$, the optimal keyframe set should have the minimal SRE; therefore, the keyframe set is identified as

$$\{r_1, r_2, \ldots, r_{\mathbf{k}}\} = \arg\min_{r_i}\{\mathcal{E}(\mathbf{V}, \mathcal{R}), 1 \leq r_i \leq \mathbf{n}\} r_{\mathbf{k}} <= \mathbf{n}\} \tag{17}$$

It is crucial in this approach that we have a correct computational model which mimics the frame interpolation function as performed by humans. However, in order to find a practical solution, certain assumptions are placed on the frame interpolation function $\mathcal{I}(.)$ and the frame difference measure $\mathcal{D}(.)$. In Hanjalic et al. [1998], Liu and Kender [2002a], and Lee and Kim [2002, 2003], each keyframe f_{r_i} represents a temporal range $[b_i, b_{i+1} - 1]$ of the video sequence, and the frame interpolation function can be written as (also see Figure 3)

$$\mathcal{I}(i, \mathcal{R}) = f_{r_t} \iff b_t \leq i < b_{t+1}. \tag{18}$$

The sequence reconstruction error function $\mathcal{E}(\mathbf{V}, \mathcal{R})$ can be rewritten as

$$\mathcal{E}(\mathbf{V}, \mathcal{R}) = \sum_{i=1}^{\mathbf{n}} \mathcal{D}(f_i, \mathcal{I}(i, \mathcal{R})) = \sum_{i=1}^{\mathbf{k}} \sum_{j=b_i}^{b_{i+1}-1} \mathcal{D}(f_j, f_{r_i}), \tag{19}$$

which is illustrated by the shaded region in Figure 3, where the absolute difference metric of a certain feature is used to measure the difference between two images.

As we can recall from Section 3.4.2 on equal temporal variance approach, in Lee and Kim [2002], b_i can be calculated without knowledge of keyframe positions, which means keyframes are selected independently of one another, assuming that the previous keyframe does not influence the position of the current one. Although this method selects the locally optimal keyframe, the computed break-points are not optimal. In order to obtain an optimal solution, at least locally, the computation of break-points and keyframes needs to be integrated. Lee and Kim [2003] therefore introduce an iterative procedure that selects a predetermined number of keyframes, while reducing the shot reconstruction error as much as possible and making the positions of keyframes and break-points locally optimal simultaneously. A similar solution is proposed in Liu and Kender [2002a], which, however, employs a heap-based structure to improve the efficiency.

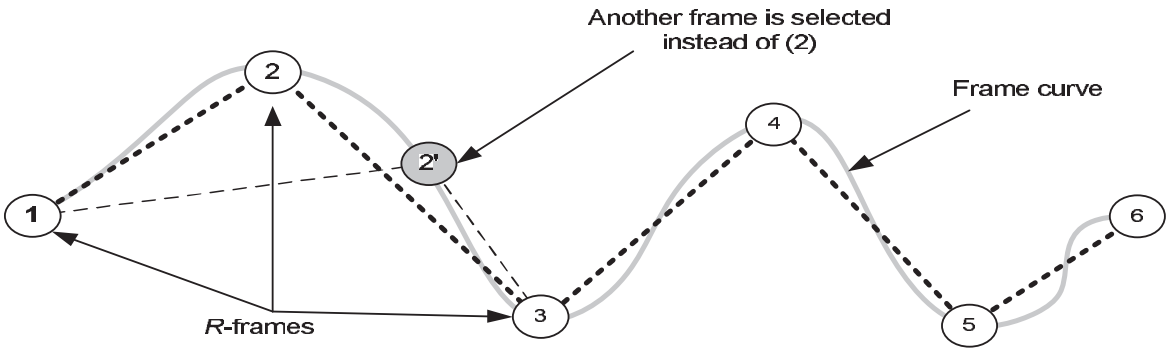

Fig. 4. Illustration of the curve simplification method.

In Hanjalic et al. [1998], each frame is represented by the cumulative activity level from the beginning of the shot, which is a monotonically increasing function. Thus, the shot distortion error is minimized only if r_i and f_{b_i} are located in the middle of their representative range (b_i, b_{i+1}) and $(f_{r_i} + f_{r_{i+1}})$, and these two conditions are sufficient for computing the final break-point $b_{\mathbf{k}+1}$, given the initial break point b_2. Based on the mismatch between $b_{\mathbf{k}+1}$ and $\mathbf{n}$, the initial value of b_2 is adjusted until the closest match is found.

Li et al. [2004] prove that an optimal solution can be found via dynamic programming if we assume that $b_i = r_i$, that is, each keyframe only represents frames that follow, as then, there is a suboptimal structure within the keyframe set. Then, assuming that the optimal solution is found for any size of keyframe set, a bisection search algorithm can be used to find the optimal keyframe set, given a fidelity constraint.

As an alternative to the aforementioned methods of using one single keyframe to interpolate others frames, two adjacent keyframes can be used, for example, the inertia-based frame interpolation algorithm in Liu et al. [2004].

3.4.7 *Curve Simplification.* The curve simplification approach is related to clustering-based methods in that it also treats each frame in the video sequence as a point in a multidimensional feature space. However, rather than looking for clusters, this approach searches for a set of points such that removal of the remaining points least changes the shape of the curve connecting all points through their temporal ordering. In this respect, it is thus related to sequence reconstruction error methods. The difference is that points on the curve are not necessarily equally spaced by the frame index, and an explicit modeling of the error between the final curve and original frame trajectory is not required. Instead, this approach uses standard curve simplification algorithms such as the binary curve splitting algorithm [DeMenthon et al. 1998], and discrete contour evolution [Latecki et al. 2001; Calic and Izquierdo 2002].

Figure 4 illustrates this approach. In the figure, six optimal keyframes are selected from a curve simplification method. If keyframe (2) is to be replaced by (2′), the new keyframe set does not approximate the frame curve as well as the original set, and is therefore less representative of the whole curve. Like sufficient content change, equal temporal variance, and shot reconstruction error methods, the keyframes produced by the curve simplification approach divide the video sequence into units of contiguous frames, hence preserving the temporal order. However, it is generally not a method of choice, as it is computationally expensive, yet not optimal.

Note that some of the methods described earlier can also be loosely classified as belonging to the curve simplification approach. For example, Hanjalic et al. [1998] try to minimize the loss associated

with approximating the area under the curve of cumulative activity. Divakaran et al. [2002] propose that the keyframe positions should divide the cumulative activity curve equally. Similarly, Sun and Kankanhalli [2000] suggest that keyframe positions should create similar unit *changes*, in which unit change is defined as the histogram difference between the first and last frames of a segment between two consecutive keyframes.

3.4.8 *"Interesting" Events.* In contrast to the preceding approaches, which focus on maximizing the extent and balance of visual coverage of the keyframe set, methods based on interesting events attempt to identify frames that are semantically important. Most work in this category assumes an association between the "interestingness" of a frame, and the motion patterns around the frame, as well as its content characteristics (e.g., containing human faces or having high spatial complexity).

In Liu et al. [2003], the extraction of keyframes is based on the motion patterns of a shot. The authors develop a triangular model of perceived motion energy, in which a video shot is segmented into subsegments of consecutive motion patterns in terms of accelerations and decelerations. They propose selection of the frame at the top vertices of triangles, which is the turning point from accelerating to decelerating motion, as keyframes. The justification is that the turning point of motion speed usually represents the most salient point of an action, and the movement within the acceleration and deceleration process can be inferred. Similarly, Han and Kweon [2005] extract keyframes at the curvature points of the 2D curve depicting the overall camera motion. However, their model incorporates not only the magnitude, but also the direction, of motion.

Dufaux [2000] propose a two-step approach to the special problem of extracting one single keyframe from a video sequence. First, shot selection is carried out based on criteria such as long shot length, high motion, spatial activity, and the likeliness to include people. Once the most representative shot of the sequence is selected, Dufaux [2000] suggests that the keyframe should be selected for the representative shot based on a low motion activity (to avoid blurring and video coding artifacts), a high spatial activity, and the high likelihood of including people.

The extraction of keyframes in Liu and Kender [2002c] is strongly dependent on the underlying data domain of instructional videos and focuses on one particular scene type: that of handwritten lecture notes. In particular, the instructional content and hence the importance of a frame is measured based on the number of ink pixels in the frame. A frame is selected as keyframes if it is a clean frame, that is, with small content variance within a temporal window, and is largely different to other clean frames.

Observing that users have a similar tendency in selecting a representative frame, Kang and Hua [2005] is the first paper that aims at learning the representative frames of a shot based on a set of visual descriptors (frame quality, content dominant, attention measurement). They first segment a shot into subshots of different classes, depending on camera motion properties, to be presented by one keyframe. From representative frames selected by users, a GMM is learned for each class of subshots, which is then used to rank the representativeness of new frames.

Although these techniques may work well for some specific experimental settings, the problem with these methods is that they strongly rely on heuristic rules drawn from empirical observations on a limited dataset. The complexity and irregularity in motion patterns of a shot may also render these solutions ineffective outside of the tested sequences or specified domain.

3.5 Keyframe Visualization Methods and the Mosaic-Based Approach

For video browsing purposes, once keyframes are extracted, they need to be presented to the user in some way that allows the user to grasp the content quickly. The two most common approaches are storyboard display and dynamic slideshow. Although screen space is an issue with the storyboard display, along with the additional time required for eye movement and user interface activities such as

scrolling, the user study in Komlodi and Marchionini [1998] found that it is still the preferred method. As for the dynamic slideshow, the user has no control over the viewing rate and may fail to comprehend the meaning and context-linking between keyframes. When extracted at different fidelity levels, it is possible to present keyframes in the layered/hierarchical manner [Parshin and Chen 2000; Sull et al. 2001]. At the top level, a single keyframe represents the entire video skip, and at the lowest level, the full listing of keyframes is shown with each camera shot being associated with one or more keyframes. However, all these approaches assume equal contribution of keyframes and image regions to the user's understanding of the video, and except for the temporal ordering, they express no other relationship between keyframes.

The purpose of alternative keyframe visualization techniques is to insert extra semantics not readily recognized with these two display forms (e.g., by establishing the relationship between keyframes) and/or to efficiently utilize the screen space. Achieving these goals requires an analysis of structural or semantic associations between keyframes and/or the availability of additional descriptors, for example, an importance score, regarding each keyframe or certain image regions within it.

Yeung and Leo [1997] propose a pictorial summary of a video sequence which consists of a set of video posters, each representing a scene in the sequence. Keyframes are extracted via a clustering-based method and assigned dominance scores based on the total duration of clusters. The video poster itself is a juxtaposition of keyframe images displayed in different sizes (which may overlap) according to a set of designed layout patterns (e.g., one predefined layout for typical dialog scenes) according to the dominance scores. Uchiashi et al. [1999] describe a similar representation called Video Manga. However, in their work, the entire video sequence is considered one single scene, and their frame-level clustering means shot extraction is not required. The Video Manga approach also assigns different image sizes to keyframes according to their dominance/importance; however, their layout algorithm maintains the temporal order of keyframes and does not utilize predefined layout patterns or allow overlapping images. This layout algorithm is further improved upon in terms of computational cost and space efficiency in Girgensohn [2003]. Recently, Chiu et al. [2004] have devised a static video visualization concept called *stained glass*. For each shot, dominant 3D regions (along the temporal axis) of high activity are first extracted and the cropped image of the first frame of the region is identified as representative for the shot and to be included in the visualization. A Voronoi layout and filling algorithm is used to expand and scale selected region images to generate the stained-glass-like visualization. Promising results are reported for staff meeting videos. However, it is unclear how much the effectiveness of the visualization depends on the accurate extraction of dominant regions.

The scene transition graph proposed by Yeung and Leo [1997] is an example of visualization methods that incorporate extra semantics not readily recognized in the conventional storyboard display. Each keyframe represents a cluster of shots, and links between them suggest the flow of the story. This representation is extended in Truong et al. [2004], who focus on unfolding various editing patterns in a movie scene.

A significant drawback of selecting a limited number of keyframes to represent the whole scene is that this cannot capture all the details of the entire scene-setting in the presence of camera movements such as pans, tilts, and rolls. In addition, it is not easy to identify the spatial relation between keyframes of a shot. When such details are required, a mosaic-based approach would be a sensible choice. A mosaic image (also called an *image sprite* [Lee et al. 1997], *salient still* [Teodosio and Bender 1993] and *panoramic image* [Taniguchi et al. 1997]) is an image synthesized from the frame sequence of a shot that shows all static components of the scene depicted by the shot. It can be considered as a visualization of keyframes in which every frame is selected as keyframes. The mosaic image of a shot is constructed by wrapping all frame images to a common coordinate system using the camera parameters estimated between two successive frames. Tonomura et al. [1993] propose a mosaic-related visualization

concept called the VideoSpaceIcon. However, instead of constructing a 2D synthesized image, estimated camera parameters are used to scale and position each frame on an axis, thereby constructing a 3D representation of the scene.

Although some success is reported in the literature (e.g., basketball and sitcom browsing in Aner and Kender [2002]), the applicability of the mosaic-based approach to real-world videos suffers greatly from its strict requirement on accurate extraction of all camera parameters and correct frame alignment. This is only possible for long-distance shots with a low level of background/foreground change, or when movement parameters are available from mechanical operations of the camera system (e.g., PZT cameras). Taniguchi et al. [1997] partially overcome this problem by constructing mosaics only when long smooth camera pan and tilt operations are detected, otherwise the first frame of the shot is used to represent the shot. They also propose a layout algorithm to pack the mosaic images of irregular shapes in a space-efficient manner. In addition, the mosaic approach is essentially shot-based and does not scale up to longer video sequences. Aner and Kender [2002] therefore propose a method for clustering the mosaic images representative for a scene (including the establishing shot, and shots with pan and zoom) to identify a set of physical settings of a sitcom episode. The mosaic image that has the best average match to the rest of the mosaics in that cluster is selected as the representative mosaic of that particular physical setting.

Another drawback of a conventional mosaic-based approach is that the generated mosaic only depicts the static background scene, and ignores dynamic components (e.g., foreground objects) of the scene. Irani et al. [1998] address this problem by proposing the *synopsis* mosaic, which is constructed by overlaying the object trajectories over the conventional mosaic image. A similar method is reported by Pope et al. [1998].

4. VIDEO SKIMS

Video skim generation is a relatively new research area and often requires high-level content analysis. The simplest method for generating video skims is uniform subsampling, which extracts fixed-duration excerpts of the original video at fixed intervals. The "fast forward" method which simply increases the frame rate across the whole video content, either uniform or adaptive to the content dynamics, is also referred to as video skimming [Omoigui et al. 1999; Peker et al. 2001; Divakaran et al. 2003; Peker and Divakaran 2004]. This approach, however, would seriously degrade coherence, causing discomfort to the viewer. In addition, it does not follow the skim definition as outlined in Eq. (2). This review focuses exclusively on video skimming works that aim at retaining the same video frame rate by selecting only "relevant" or "important" video segments. In this way, given knowledge about the keyframes, one simple and straightforward method for skim generation is to include the neighborhood of keyframes to form continuous segments and join them together, as done in Wu et al. [2000] and Ouyang et al. [2003]. However, video skims generated in this fashion are still hard to comprehend, and therefore, more effective and complex techniques have been developed independently of the keyframe extraction process, which shall be reviewed here.

Figure 5 shows our characterization of skim generation techniques proposed in the literature. As can be seen from this figure, there are five major aspects of a skim generation technique: its overall process, the nature of skim length, the genre/domain of the video sequence to be summarized, the underlying mechanism, and the features used. In the coming subsections, we shall describe these aspects in detail and explain how they are addressed in existing works.

4.1 Skim Length

As with keyframe extraction, the length of a video skim can be specified as *a priori* or left unknown until the end of the extraction process (i.e., *a posteriori*).

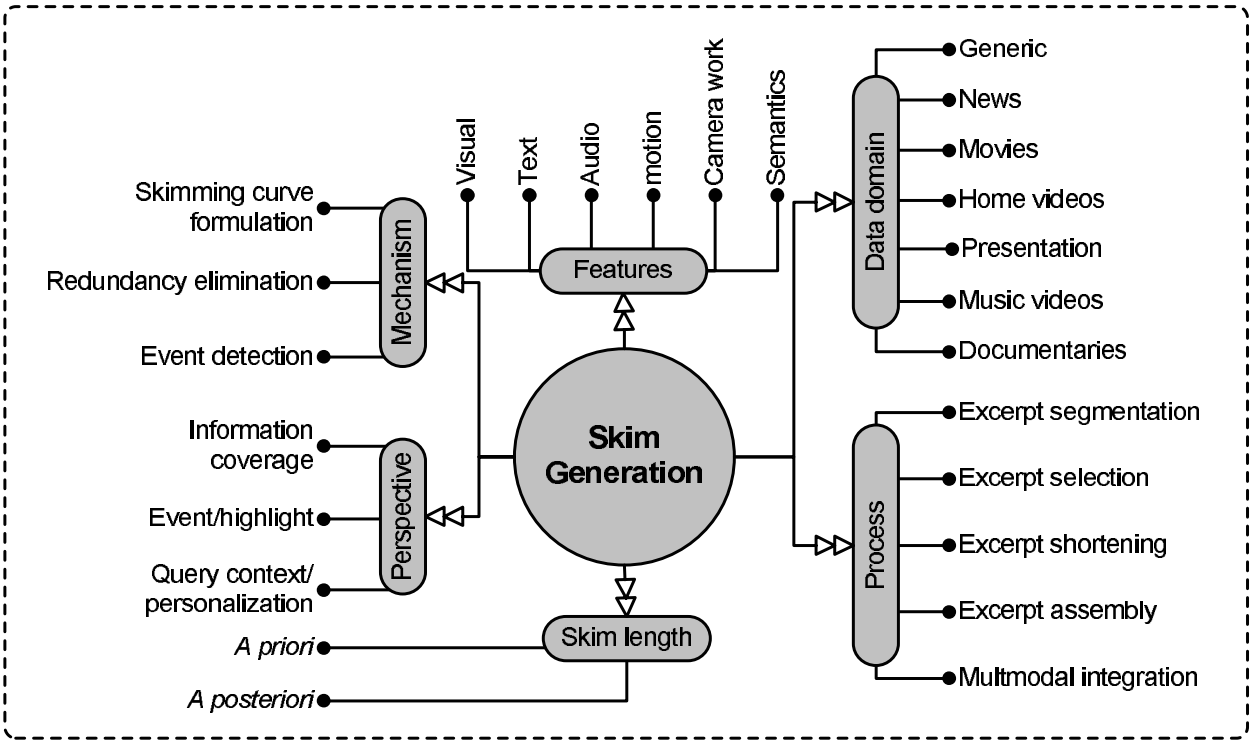

Fig. 5. Attributes of video skimming techniques.

4.1.1 *A Priori.* The length of the expected skim is specified either as a time duration or a ratio over the length l of the source video. We can formalize the optimal skim generation problem as finding $\{\mathbf{E}_1, \ldots, \mathbf{E_k}\}$ such that the skim produced is least different to the original sequence with respect to a certain summarization perspective ρ:

$$\mathcal{K} = \arg\min_{\mathbf{E}_i}\{\mathcal{D}(\mathbf{E}_{i_1} \odot \mathbf{E}_{i_2} \odot \ldots \odot \mathbf{E}_{i_\mathbf{k}}, \mathbf{V}, \rho) \mid \sum_{i=1}^{\mathbf{k}} |\mathbf{E}_i| = l, \mathbf{E}_i \sqsubset \mathbf{V}\}, \tag{20}$$

where $\mathbf{E}_i$ is the ith excerpt to be included in the skim, $\odot$ is the excerpt assembly and integration operation, and $\mathcal{D}$ is a similarity measure. For example, if perspective ρ is information/coverage, then the problem is to find and combine the portions of original clips that give the most amount of information about the original video. He et al. [1999] and Ma et al. [2002] are among video skimming works that require the length of skim to be specified by the user.

4.1.2 *A Posteriori.* The length of skim is determined by characteristics of the underlying video. It is the minimal number (or total length) of excerpts required for the skim so that the generated skim is not much different to the source material with respect to the selected perspective. Formally, the video skim $\mathcal{K}$ is generated as

$$\mathcal{K} = \arg\min_{\mathbf{E}_i} \left\{ \sum_{i=1}^{\mathbf{k}} |\mathbf{E}_i| \mid \mathcal{D}(\mathbf{E}_{i_1} \odot \mathbf{E}_{i_2} \odot \ldots \odot \mathbf{E}_{i_\mathbf{k}}, \mathbf{V}, \rho) < \varepsilon \right\}, \tag{21}$$

where ε is the tolerance value. For example, if the perspective ρ is the event, then the problem is to create the shortest summary that allows the user to observe all interesting events. Babaguchi [2001] and Sundaram and Chang [2001] are examples of work that let the length of the generated skim be determined internally by underlying attributes of the input video.

For techniques that utilize the hierarchical structure of a video document (e.g., sequence-scene-shot), the length of skim is distributed downward in the hierarchy, and the skimming of each unit can be performed independently. For example, in Lu et al. [2004a], the skim length for each movie scene is proportional to its content entropy, which is derived from the total scene length and the collective lengths of each cluster formed in the scene.

4.2 Target Video Domain

Most proposed techniques for video skim generation are domain-dependent, that is, they exploit features available to a particular kind of video. In the literature, skims have been produced for the following kinds of video: sports [Babaguchi 2001; Babaguchi et al. 2000; Marlow et al. 2002; Xiong et al. 2003a; Ariki et al. 2003; Hanjalic 2003; Coldefy and Bouthemy 2004; Coldefy et al. 2004], news [Smith and Kanade 1998; Christel et al. 1998], documentaries [Yu et al. 2003], movies [Pfeiffer et al. 1996; Lienhart et al. 1997; Hanjalic et al. 1999; Sundaram and Chang 2001; 2002], home videos [Lienhart 1997; Zhao et al. 2003; Yu et al. 2003], and lecture recordings [He et al. 1999]. A technique is classified as *generic* if it can be used across many video domains. For example, video skimming based on the detection of sound effects such as laugher, cheering, and applause (as proposed in Cai et al. [2003]) can be applied to sports and entertainment videos. Also, techniques relying on redundancy elimination are often applicable to many kinds of video [Cooper and Foote 2002; Gong and Liu 2003; Gong 2003].

4.3 Skim Generation Process

We identify a generic five-step process for generating video skims automatically: excerpt segmentation, excerpt selection, excerpt shortening, multimodal integration, and excerpt assembly. Note that the term *excerpt* here refers to a segment of video, be it shot, scene, or event spanning across a number of shots, etc. Although, in practice, a skim generation technique may skip certain steps or combine them in different variations, the principle and essence remains as described next for all video skimming works.

4.3.1 *Excerpt Segmentation.* This step segments the whole video sequence into separate units, some of which will be included in the skim. However, the segmentation of the video sequence into anonymous segments, such as shots, is often considered an *a priori* process, rather than as part of the skim generation. Segment boundaries in a complete textual source such as the closed caption are simply punctuation marks, whilst the segmentation of speech is done in Taskiran et al. [2001] by detecting pauses in speech using the timecode of the extracted transcript. For techniques based on the detection of important/interesting events (e.g., Rui et al. [2000] and Ariki et al. [2003]), excerpt segmentation is essentially the process of dividing the video sequence into event and nonevent segments. As for other approaches, Peyrard and Bouthemy [2003] perform sequential segmentation of the video by detecting changes in the dominant image motion. The factorization of the self-similarity matrix employed in Cooper and Foote [2002] can also be considered a form of excerpt segmentation.

4.3.2 *Excerpt Selection.* The next step is to select the excerpts to be included in the skim (not yet shortening). The coverage of generated skims will depend on the technique used for excerpt selection and will also influence the context and coherence level of the skim. Excerpt selection can also be conducted by first clustering shots and selecting only that one shot from the group with the longest duration [Gong and Liu 2003]. For event-based approaches, excerpt selection is simply the process of picking up relevant events to be included in the skims [Ariki et al. 2003; Peyrard and Bouthemy 2003]. Sometimes the skim

length and other factors must be taken into account in selecting events, in which case a measure of the event significance/impact is often required. For example, Babaguchi et al. [2000] select events in decreasing order of impact until the skim reaches the desired length. In their subsequent work, they select events by traversing through a 2D table of event significance and event shot type (i.e., live-shot motion, live-shot still, replay shot, etc.). Excerpt selection can also be done recursively, as shown in Ngo et al. [2003]. First, they form scenes, and clusters, of shots. Motion-based attention is computed for shots, clusters, and scenes, together with prior probability of occurrence of each cluster. Based on these two attributes, the elimination of video segments is done at a scene, cluster, and shot level to determine which shots are to be included in the skim. In Miura et al. [2003], excerpt selection is simply the process of filtering out face shots from the abstraction of cooking videos, as these shots contain little cooking information.

4.3.3 *Excerpt Shortening.* A correct shortening procedure will ensure that the excerpt is concise, without a notable loss of information. However, shortening an excerpt runs the risk of creating inappropriate cut points (e.g., in the middle of a speech sentence), reducing overall coherence and irritating the viewer. The simplest method for shortening an excerpt is to select a predetermined portion of the excerpt. For example, Gong and Liu [2001, 2003], and Lee et al. [2004] assign only a specified time length to each selected excerpt (shot in this case). Similarly, He et al. [1999] only take the appropriate duration segment immediately after each slide transition in skimming lecture presentations. Cooper and Foote [2002] use a self-similarity matrix to select that contiguous portion of the excerpt which is most similar to the whole excerpt. Sundaram and Chang [2001] model the distribution of comprehension time, given the visual complexity of a shot. A shot can be shortened to the upper bound of its comprehension time. Ma et al. [2002] shorten each shot by identifying keyframes from the attention curve and use only those segments around these keyframes. In Miura et al. [2003], only those portions of the selected shots for a skim of cooking videos that correspond to cooking motion or the appearance of food are selected. If selected excerpts can be classified into different semantic classes (e.g., speech, music, laughter, etc.), then their durations can be shortened proportionally to the user preference for these classes [Zhao et al. 2003]. If the selected excerpt is an event panning across multiple shots, the shortening can be done by removing silent shots, as these do not contain important messages [Li et al. 2003]. However, if the summarization perspective is highlighting, this step is generally omitted, allowing the user full comprehension of highlighted parts of the video sequence. Instead of shortening excerpts, Li et al. [2003] try to compress them by dropping frames and modifying the audio signal to an acceptable limit.

4.3.4 *Multimodal Integration and Excerpt Assembly.* The generated excerpts discussed thus far are often single-modal (audio, image, or textual). The purpose of this step is to insert the missing modality, realign excerpt boundaries, and combine all excerpts into the final skim. Multimodal integration, if done correctly, can enhance the coverage, context, and coherence of the produced skim.

Video skims can be classified into two types based on the integration of audio and video information: modal synchronization and modal asynchronization. In the former, the audio stream and visual stream are synchronized according to the timeline of the original video sequence. This type of skim is essential to video programs such as movies, dramas, and talk shows, since what we hear from the audio track directly corresponds to what we see on the screen. The common timecode for the selected and shortened excerpts can be generated based on visual information only (i.e., visual-centric), audio information only (i.e., audio-centric), textual information only (i.e., textual-centric) or a combination of these. When skim excerpts are generated separately for individual modalities, they can be integrated to produce the final modal synchronization skim by employing the OR (or AND) operation on the timecode, as done in [Erol et al. 2003], which generates a skim consisting of all video segments that are contained either

in audio-centric or visual-centric excerpts. The OR operation is also used in Agnihotri et al. [2004]. To improve smoothness of the final skim, Erol et al. [2003] also merge segments that are temporarily close to each other. Since the textual stream, such as closed caption, often contains spelling errors and is not aligned to the speech signal, an alignment algorithm is also required to map the timecode to the words in the textual stream [Kim et al. 2004]. The synchronization is ensured by selecting segments from all streams according to the common timecode.

For video programs such as news and documentaries, a modal asynchronization skim can be useful in maximizing the information coverage. Video skims of this kind can also be visual-centric, audio-centric, or textual-centric. However, the missing stream is constructed by selecting semantically appropriate segments and inserting them into the final skim. For example, in Smith and Kanade [1998], visual elements are inserted into the audio-centric skim based on a set of heuristic rules regarding object/camera motion, the human face, and text captions. For news sequences, sometimes it is better to retain only the anchor audio (allowing the user full comprehension of the story headline) and to overlay this on the visual summary so as to produce the final skim [Lie and Lai 2004; Xie et al. 2004]. Gong and Liu [2003], on the other hand, integrate visual-centric and audio-centric skim excerpts via a bipartite graph-based alignment algorithm.

For modal synchronization skims, the excerpt boundaries generated using visual information may break the coherence of audio stream. The purpose of the aligning operation is to adjust the boundaries of originally extracted excerpts so that the flow, coherence and context in all information modals are maintained. Audio smoothness is maintained by ensuring that it is not interrupted in the middle of a sentence, thus, the original timecode of skim excerpts are often aligned to speech sentence boundaries. If the audio transcript is available, the timecode of sentence boundaries can be extracted by locating punctuations in the transcript and aligning the transcript to actual speech [Kim et al. 2004]. Otherwise, sentence boundaries are estimated via detection of pauses in speech [Taskiran et al. 2001; Ma et al. 2002].

The most straightforward method for assembling excerpts into the final video skim is to join them in their temporal order. This is in fact employed by almost all skim generation techniques. The exception is Lienhart et al. [1997], who divide excerpts into classes {event, dialogue, filler, extracted text} and across these classes the temporal order can be broken, hopefully producing more interesting movie trailers. Lienhart et al. [1997], Lienhart [1997], and Erol et al. [2003] aim to improve the coherence/flow of the generated skim by joining clips with shot transition devices such as fade and wipe. Likewise, Pan et al. [2001] insert gradual transitions between highlight segments of sports videos. However, these works do not provide a systematic evaluation of user satisfaction with alternative excerpt assembly operations. Russell [2000] attempts to improve the coherence and flow of video presentation abstracts by employing a set of summary design patterns (SDPs). Detected segments and their semantic labels are used to identify the appropriate SDP and these segments are then mapped to the selected SDP to render the summary.

4.4 Preserved Perspective

In order to generate a skim, we must decide which perspective of the video document needs to be preserved as much as possible. Different summarization perspectives create different forms of video skims and are suitable for different target video domains and applications. Our review of the literature reveals the following three broad perspectives: information/coverage, important/interesting events, and query context/personalization.

4.4.1 *Information Coverage.* Sometimes the term *video skim* exclusively means the dynamic video abstracts of this kind. They primarily aim at providing an impression of the entire original video

by eliminating redundancy and shortening segments, without severely affecting the user's comprehension. They therefore are mostly applicable to information-centric videos (e.g., broadcast news and instructional videos) in which users are interested in the content and meaning of the video as a whole, rather a few particular moments. Preserving information and content coverage in video skimming is addressed in Lienhart [1997], He et al. [1999], Hanjalic et al. [1999], Sundaram and Chang [2001, 2002], Ma et al. [2002], Gong and Liu [2003], and Gong [2003].

4.4.2 *Important/Interesting Events.* Often appearing under the name *video highlights* [Hanjalic and Zhang 1999], the purpose of these video skims is to preserve interesting or important events in the video. They are most applicable to sports videos, where the concept of the "interestingness" of a video excerpt can be defined, modeled, and computationally extracted. In practice, a video excerpt is identified as important, interesting, or exciting if it:

—is an event with a specific semantic label, for example, the scoring of a goal in soccer videos [Assfalg et al. 2003];

—has specific temporal and spatial attributes, for example, occurring around the goal mouth of a soccer game [Wan and Xu 2004a];

—evokes certain reactions from the live audience, for example, applause and cheering [Xiong et al. 2003a];

—evokes certain reactions from the narrator/commentator, such as, excited speech [Marlow et al. 2002; Coldefy and Bouthemy 2004];

—evokes certain reactions from the content producer, for example, the cutting rate, score caption, and replay sequence [Pan et al. 2001; Tjondronegoro et al. 2004b];

—is used in several video sequences and by different content producers, such as, repeated footages in news programs [Bagga et al. 2002];

—captures the attention of the viewer, for example, high motion and colorful [Ma et al. 2002];

—has specific occurring patterns, for example, rare and unusual [Xiong et al. 2003a; Radhakrishnan et al. 2004];

—has specific viewing patterns, such as those, viewed most often by the user [Masumitsu and Echigo 2000; Yu et al. 2003]; or

—is associated with psychological patterns of the filming person captured simultaneously by an external device, for example, high magnitude of α-brainwaves [Aizawa et al. 2001].

The preceding shows that all subjects involved in the capture, production, and consumption of the video contribute to the judgment of how interesting a video segment/event is, all of which can be exploited to construct a computational model for video skim generation. However, a major problem still exists in finding the correct integration model and also in identifying the boundaries of detected events so as to ensure the conciseness of the skim, while preserving sufficient event context.

4.4.3 *Query Context and Personalization.* Video skims of this kind are generated according to the user specification, either as a query or a personalization profile. They simplify the problematic process of deciding which events to select for highlighting by retaining those portions of the video that are most relevant to the query or user information needs. In Christel et al. [1999], skim excerpts are extracted by matching the query and the audio transcript. Personalization is often implemented by having the user specify preference weights over a set of features (including event type) associated with candidate video excerpts. For example, relevant features in Lu et al. [2003] are {human face, speech, camera zoom, caption text} for news video and {gunshot/explosion, loud voice, face, fire color} for movies. For

abstracting home videos, in Zhao et al. [2003], weights are required for temporal exclusive events such as {speech, laughter, song, applause, scream, motion, blur}, whilst for movies, Li et al. [2003] propose a set of six features {music ratio, speech ratio, sound loudness, present cast, action level, and theme topic} extracted from any story event (two-person dialog, multiperson dialog, and hybrid scene). For sports videos, favorite teams, favorite players, and events of interest can be personalized [Babaguchi et al. 2000]. Jaimes et al. [2002], on the other hand, learn user preference from a set of highlight examples. Rather than having the user specify preferences, Agnihotri et al. [2005] attempt to produce video summaries that are adaptive to the personality trait of the user, for example, male/famele, introvert/extrovert.

4.5 Underlying Mechanisms

The fundamental mechanism used to generate a video skim is often dependent on the perspective that the skim aims to preserve. For example, techniques focusing on detecting interesting events will not do well in preserving the coverage aspect of the video. We identify the following three fundamental and nonexclusive mechanisms extensively used in existing video skimming works: redundancy elimination, event detection, and skimming curve formulation.

4.5.1 *Redundancy Elimination.* Here, by redundancy, we mean the portion of video sequences that contains relatively known information. Excerpt shortening is an essential part of the redundancy elimination process in the sense that it retains only enough data in each selected excerpt for comprehension [Sundaram and Chang 2001, 2002]. Cooper and Foote [2002] remove redundancy in an excerpt by selecting only continuous frames that maximize the average similarity to the entire sequence, while Ma et al. [2002] only select interesting portions of an excerpt. Redundancy can also be done in the excerpt selection process via clustering, as in Hanjalic and Zhang [1999], Gong and Liu [2001], Gong [2003], and Ngo et al. [2003], in which the maximum of one shot is retained from a cluster of visually similar shots. Lu et al. [2004a, 2004b], on the other hand, eliminate redundancy via an optimization method related to the minimal correlation approach for keyframe extraction described in Section 3.4.5. Specifically, their work postulates that the optimal set of excerpts should have minimal total similarity between successive excerpts, and the optimization problem is then solved via the dynamic programming method. On the audio stream, redundancy elimination can be performed differently by removing sequences with background noise and silence. As the redundancy elimination process uses information from only one channel, it faces a crucial problem: A segment identified as redundant in one channel may contain new information in another.

4.5.2 *Event/Highlight Detection.* For event-preserving video skims, mechanisms for event extraction are often required which need to not only identify the occurrence of an event, but also to locate the boundary of the event context. Chang et al. [2002] develop separate HMM models to detect baseball events, including "home run," "catch," "hit," and "infield play" to include in the highlights. Their models require the classification of a baseball shot into one of seven classes: {pitch view, catch overview, catch closeup, running overview, running closeup, audience view, touch-base closeup}. Dagtas and Abdel-Mottaleb [2004] detect highlight events in sports videos via two methods: locate the co-occurrence of keywords such as "goal" and "touch-up" with high audio energy; and detect the transition between a sequence of frames of large grassy areas (game action) to a sequence of frames that do not include large grassy areas (closeups). Babaguchi [2001] detects score-changing events (touchdown, field goal, extra point) in an American football game by analyzing closed-caption streams and shot template matching. In Peyrard and Bouthemy [2003], sports events are detected by relying only on motion features. Sequential segmentation of the video is first carried out by detecting changes in dominant image motion. For each segment, they adopt statistical representation of the residual motion content to classify into

predetermined classes (e.g., pole vault, interview shots, large views of the stadium, etc., for an athletics video) according to the maximum likelihood principle. Assfalg et al. [2003] detect soccer events using temporal logic models. Each model depicts the relationship between playfield zones, ball motion, and player positions for a specific event.

Instead of detecting specific game events, highlight segments can be identified by detecting the audio, visual, or cinematic events that would be triggered by important game events. In sports videos, interesting game events are often replayed. A replay sequence can be a slow-motion replay, an exact replay, or an alternative replay (i.e., showing the event from different camera angles at normal speed). Most work in replay detection focuses on slow-motion replay [Gu et al. 1999; Pan et al. 2001; Ouyang et al. 2003; Tjondronegoro et al. 2004a], whilst the detection of exact replay is also investigated in Gu et al. [1999]. Xiong et al. [2003a] use only audio features and the entropic prior HMM, together with pre- and postprocessing techniques, to generate sports highlights by detecting applause and cheering. This work is extended in Xiong et al. [2004], in which Gaussian mixture models (GMMs) and minimum length description length criterion are used to detect audio events. These are used together with motion activity to identify candidate highlight segments. Rui et al. [2000] employ an integrated approach where thrilling baseball segments are assumed to be highly correlated with the announcer's excited speech and most of the enthrallment of the game is assumed to occur right after a baseball pitch and hit. They propose techniques to reliably detect excited human speech and baseball hits, and then fuse them in a probabilistic framework. Similarly, excited speech sequences are extracted as candidate highlight segments in summarizing soccer videos in Coldefy and Bouthemy [2004] and false alarms caused by commercial sequences are filtered via the detection of dominant camera motion patterns and green color regions. Wang et al. [2004] simply detect "plopping" sound effects in diving videos to identify diving action shots, which will be used as the highlight.

Rather than constructing a particular model for highlight-related events, Radhakrishnan et al. [2004] observe that in sports videos, interesting events often happen sparsely in a background of usual and uninteresting events. Therefore, a video skim can be constructed by detecting outliers or "unusual" events by constructing a statistical model of the background process. However, this method is only applicable when variation of the background process is low.

4.5.3 *Skimming Curve Formulation.* This approach proceeds by computing a score that directly associates each base unit with its likelihood to be included in the skim with respect to perspective ρ. For example, if the user would like to see interesting events in the skim, the feature should reflect how interesting a base unit is. The base unit is often primitive and modal-dependent: a frame or shot for visual-centric skims, a fixed-size clip for audio-centric skims, and a word or sentence for textual-centric skims. In order to improve coherence, the base unit can be a high-level structural unit identified by the excerpt segmentation step. For example, Taskiran et al. [2001] use the segment between two pauses in speech to ensure that skim excerpts do not begin and end in the middle of a phrase, whilst Li et al. [2003] define base units as story events spanning across a number of shots: {two-speaker dialog, multispeaker dialog, hybrid event}. Figure 6 shows a straightforward process for skim generation based on the perspective curve. In this process, excerpt segmentation is done by setting a threshold on the perspective curve, and excerpts with perspective values above this threshold are selected and assembled into the final skim. Excerpts can also be determined by searching for local maxima if the interestingness is treated as a temporally relative concept [Ma et al. 2002; Xiong et al. 2003a].

If the base unit is time-based, namely, frame or fixed-size segments, a constraint on skim duration can be satisfied easily by extracting units with the highest scores until the total duration reaches the constraint. It is less straightforward if base units have a different temporal duration, such as, shots or speech sentences. In this case, the problem of excerpt/unit selection can be seen as the 0-1 KNAPSACK

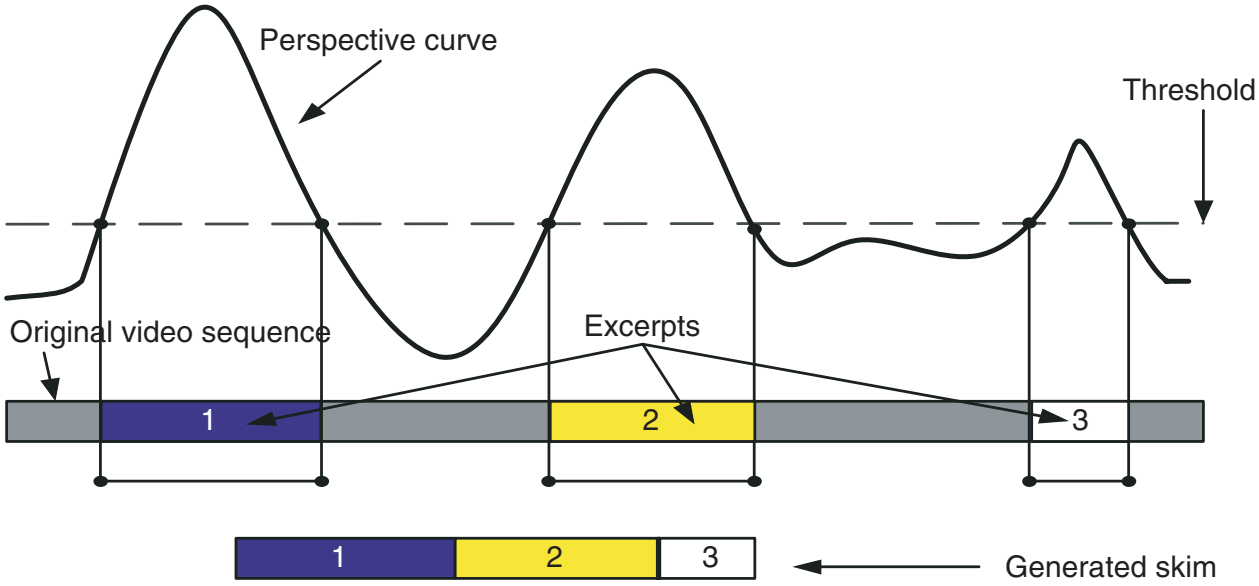

Fig. 6. Illustration of skim generation from a skimming curve.

problem: maximizing the total score subject to an upper bound on the total duration. The solution in existing skimming works is to sort each excerpt/unit according to its skimming likelihood score [Li et al. 2003; Xie et al. 2004] or efficiency (likelihood score divided by time duration) [Taskiran et al. 2001], and then employ a greedy selection algorithm.

The crux of this method is how to derive the interestingness score of the base unit. If the base unit is a sports game event, the score can be associated with the type of event and/or the volume and duration of excited speech, applause, and cheering. If the base unit is text-based, techniques developed for text summarization can be used. For generic units, a common approach is as follows: A set of features (including event type) are defined over the base unit, and for each feature, its contribution to the interestingness of the unit is computed using generic models. The skimming curve is then formed by summing over these contributions. Personalization is implemented as prior weights of the features. For example, Hanjalic [2003] models the interestingness of video content via overall motion activity, cutting rate, and the energy contained in the audio track. In Lu et al. [2005], each shot is assigned an importance value computed from its description (manually annotated) and according to the *mutual reinforcement principle*, which basically enforces that an important term should occur in many important descriptions and that an important description should also contain many important terms. Rather than associating the interestingness of a video unit with audio/visual features, interestingness can be gauged from user interaction with the entire sequence [Masumitsu and Echigo 2000; Yu et al. 2003]. Yu et al. [2003] model the interestingness of a shot via a concept called *ShotRank*. The ShotRank of a video shot represents the probability that a viewer will visit a shot during browsing and is computed from the information contained in previous viewers' browsing logs. In Aizawa et al. [2001], the interestingness of a shot is measured by the level of interest of the person capturing the footage. In their work, the α-brainwave forms the skimming curve and thresholding can be used to extract interesting segments. This uses the premise that, according to psychological research, α-wave attenuation occurs when a person feels awake, interested, and excited. Instead of thresholding, Ng et al. [2002] use an SVM classifier to determine segments as either "interest" or "no interest" using the magnitude of α-wave together with visual

features. The main application of these works is the summarization of wearable videos that capture a person's life experience.

If the base unit is a frame, fixed-size segment, or shot, the generation of video skims as demonstrated in Figure 6 does not ensure coherence and balanced content coverage, as it may contain very short or visually similar excerpts. Lu et al. [2005] attempt to overcome this problem by finding a minimum set of excerpts that maximize the total importance scores and minimize the number of shots in selected excerpts which fall into the same semantic cluster. This is solved in their work via a greedy method. Similarly, in Mei et al. [2005], the skimming curve is formed for home video sequences by estimating its quality from a set of metrics, including *unstableness, jerkiness, infidelity, blurring, brightness, and orientation*. Instead of thresholding the skimming curve, they ensure the coverage of produced skims by selecting subshots uniformly distributed throughout the sequence via the formulation of excerpt selection as an optimization problem.

4.6 Features Used

It is evident from the discussion so far that various features have been used for skim generation, spanning across different categories and modalities:

—*Visual.* In sports videos, dominant colors, edges and textures and their spatial arrangement are crucial in identifying camera views and positions of play (e.g., overview shot, closeup shots of players, near goal, etc.) which in turn can be used to infer the presence of highlight segments [Chang et al. 2002; Han et al. 2002; Ariki et al. 2003; Assfalg et al. 2003; Shih and Huang 2004; Chen et al. 2004; Dagtas and Abdel-Mottaleb 2004; Tjondronegoro et al. 2004b]. In addition, visual features such as the colour histogram are often used to measure the similarity between frames/shots, which is crucial to video skimming methods based on redundancy elimination, as shown in Gong and Liu [2001, 2003], Gong [2003], Ngo et al. [2003], Lu et al. [2004a, 2004b], and Lee et al. [2004]. Ma et al. [2002] compute elements such as *color contrast, intensity contrast*, and *orientation contrast* to model the human attention level to a particular image. Pfeiffer et al. [1996] extract high-contrast scenes to include in movie summary.

—*Text.* When available, text is a very powerful feature, as the semantic concept can be extracted more easily, accurately, and efficiently compared to visual and audio data. Text can be acquired from various sources: (a) within the video stream, which includes superimposed captions, textual overlays, and graphical inserts [Pfeiffer et al. 1996; Lienhart et al. 1997; Lienhart 1997]; (b) closed captions accompanying the video stream [Han et al. 2002; Miyauchi et al. 2003; Dagtas and Abdel-Mottaleb 2004]; (c) speech recognition [Smith and Kanade 1998; Ariki et al. 2003; Gong 2003]; and (d) external transcripts (e.g. *gamestat* American football database [Babaguchi et al. 2000, 2001]). Textual data is often exploited for video skimming via the identification of keywords and index terms (e.g., "touch-up," "goal"), which are reliable indicators of sports events [Babaguchi 2001; Babaguchi et al. 2000; Han et al. 2002; Miyauchi et al. 2003; Dagtas and Abdel-Mottaleb 2004]. These terms can be specified manually or learned directly from the training data [Miyauchi et al. 2003]. Smith and Kanade [1998] and Gong [2003], on the other hand, produce audio-based summary by applying text summarization techniques such as TF-IDF, or by utilizing latent semantic analysis on the speech transcript to extract the sentences and phrases that best represent its content.

—*Audio.* This category refers to primitive audio features such as energy, envelope, Mel frequency cepstral coefficients (MFCCs), etc. For computational efficiency, features directly available in the compressed audio-bit stream, like scale factor, can also be used [Marlow et al. 2002]. Ma et al. [2002] model human attention to an audio segment based on its energy, sound type, and duration. Likewise, Marlow et al. [2002] and Dagtas and Abdel-Mottaleb [2004] show that an increase in audio energy and

amplitude alone, especially in speech band, can be associated with an increase in the interestingness of sport actions. However, the most typical application of low-level audio features is to use multiple low-level features to construct a model (e.g., HMMs, GMMs) for detecting generic sound effects associated with the excitement in most videos such as {cheer, applause, laughter, excited speech} [Han et al. 2002; Petkovic et al. 2002; Cai et al. 2003; Xiong et al. 2003a, 2003b, 2004; Coldefy and Bouthemy 2004; Coldefy et al. 2004; Sugano et al. 2004; Wang et al. 2004; Wan and Xu 2004b; Tjondronegoro et al. 2004b]. Alternatively, the sound events associated with a specific video type can be detected to identify potential highlight segments, for example, plopping sounds in diving sequences [Wang et al. 2004], hits in baseball sequences [Ariki et al. 2003] or a whistle [Tjondronegoro et al. 2004a]. Aoyagi et al. [2004] and He et al. [1999] use primitive audio features and pitch tracking to identify speech segments and their emphasis, which are the main criteria for skimming, whilst Radhakrishnan et al. [2004] detect interesting segments from sports videos in an unsupervised manner by identifying unusual sound patterns.

—*Visual dynamics.* The level of visual dynamics is often associated with the interestingness and excitement of a video segment. It can be used to detect action scenes in a movie [Pfeiffer et al. 1996] an interesting play in a sports game [Petkovic et al. 2002; Peyrard and Bouthemy 2003; Hanjalic 2003; Xiong et al. 2003a; Wang et al. 2004], or cooking actions in a cooking program [Miura et al. 2003]. It also serves as an important element in directing the viewer's attention to a particular moment in the video sequence, as modeled in the skimming curve method [Ma et al. 2002; Ngo et al. 2003]. A pattern of visual changes across successive video frames can also be useful in identifying mid-level semantic concepts such as slow-motion replay sequences [Pan et al. 2001; Kobla et al. 1999; Ekin and Tekalp 2002].

—*Camera motion.* In videos of sports games, camera parameters are strongly related to the movement of the player and ball, which exhibit different signatures for various game actions, including highlight segments [Han et al. 2002; Assfalg et al. 2003; Coldefy et al. 2004; Shih and Huang 2004]. In Ma et al. [2002], specific camera movement patterns are incorporated in the user attention model for video skimming.

—*Mid-level semantics.* This category refers to the use of other common mid-level semantic features in the video skimming process, which are extracted independently of it. In Miura et al. [2003], face shots are considered irrelevant and excluded from the abstract of cooking videos, as they contain little cooking-related information. In Lienhart [1997], they can also be used to classify movie scenes into dialog and non-dialog. Ma et al. [2002] model human attention to a shot based on the presence of faces and their sizes. Other features include logged information that shows the patterns of user access to a video document [He et al. 1999; Yu et al. 2003] and event metadata available with the video stream of soccer games encoded in Mpeg-7 format [Jaimes et al. 2002; Takahashi et al. 2005]. For videos for which the production data is available, additional information can be incorporated. For example, Erol et al. [2003] utilizes the output of the sound-source localization module for abstracting meeting videos.

5. EVALUATION METHODS FOR VIDEO ABSTRACTION TECHNIQUES

In order to advance the field, the effectiveness and/or efficiency of a new solution to a particular problem needs to be evaluated, preferably against existing methods. However, a consistent evaluation framework is seriously missing in video abstraction research, resulting in the fact that every work has its own evaluation method, often lacking the performance comparison with existing techniques. This is partly because, unlike other research areas such as object detection and recognition, evaluating the correctness of a video abstract is not a straightforward task due to the lack of an objective ground-truth. It is even

difficult for humans to decide if one video abstract is better than another, and to make matters worse, the summarization viewpoint and perspective are often application-dependent. In addition, duplicating previous work involves a large overhead, especially for video skimming techniques, whilst existing systems are often unavailable for comparative evaluations or not easy to use with the data in different formats and settings. In this section, we describe existing methods for video abstraction, which are grouped into three different categories (result description, objective metrics, and user studies) and outline suggestions for establishing a consistent evaluation framework for the field.

5.1 Result Description

This is the most popular and simple form of evaluation, as it does not involve any comparison with other techniques. Often, the proposed technique is applied to a few video sequences and the generated video abstract displayed and/or described, thus indicating whether the method adequately generated the abstract. It is also common to discuss the influence of the system parameters or visual dynamics of the video sequence on the keyframe set extracted [Zhuang et al. 1998; Hanjalic et al. 1998; Zhang et al. 2003; Yu et al. 2004]. Some works may attempt, in descriptive form, to explain and illustrate advantages of the proposed technique compared with some existing methods [Joshi et al. 1998; Vermaak et al. 2002]. We believe this form of evaluation to be inadequate; it is largely subjective and provides little experimental justification of why the proposed technique would still work outside of the few video samples described.

5.2 Objective Metrics

For keyframe extraction techniques, this metric is often the fidelity function computed from the extracted keyframe set and original frame sequence. The metric is used to compare the keyframe set generated by different techniques, or by one underlying technique, but with different parameter sets. However, the so-called objective metrics are biased toward certain summarization viewpoints or closely linked to the proposed technique. There is also no experimental justification for whether the metric maps well to human judgement regarding the quality of a keyframe set. This metric is typically the same as the criteria used for formulating the optimal keyframe set as shown in Eqs. (3) and (4). For example, Liu et al. [2004] compare the performance of their SRE-based method against Wolf [1996], Zhang et al. [1997, 1998, 2003], Tie-Yan Liu [2000], and Lee and Kim [2002] using the SRE metric (see Section 3.4.6 and the Semi-Hausdoff distance described in Section 3.4.3), and show that their method achieves better scores with each specified number of keyframes. The only metrics that would map well to human perception of the coverage and redundancy of a keyframe set are those based on the alignment of video frames to a common coordinate system, as proposed by Fauvet et al. [2004].

Chang et al. [1999] demonstrate the effectiveness of their algorithms by plotting, against the fidelity level, the number of keyframes extracted via the sufficient content change method and the optimal exhaustive search algorithm (under their maximum frame coverage framework). A similar method is employed in Liu and Kender [2002b], together with a concept called "well-distributed" keyframes. In the authors' view, a keyframe is considered well-distributed if it neither duplicates the coverage within a shot nor falls within the gradual transitions between shots. By plotting the number of well-distributed keyframes against the number of extracted keyframes, Liu and Kender [2002b] show that their algorithm is closest to the ideal curve and hence, better than three other methods based on the first frame of the shot, clustering, and frame coverage. Kim and Hwang [2002] use a ground-truth object mask to evaluate whether extracted keyframes capture important object-related events in video sequence.

In the context of videos captured in ubiquitous homes, de Silva et al. [2005] show that a set of ground-truthing keyframes can be produced by averaging the keyframe sets selected by eight subjects. They demonstrate the quality of the produced keyframe set by analyzing the temporal matching against

the ground-truth set. In a similar approach, Kang and Hua [2005] show that their technique based on GMM modeling of the representativeness produces a much better keyframe set than selecting the middle frame or that with the highest color entropy.

For skimming work based on the detection of interesting/important segments (see Section 4.5.3), performance evaluation can be carried out via common information retrieval metrics such as precision and recall, since these segments can be ground-truthed. Often, a detected event is claimed as "correct" if its temporal extension overlaps with any ground-truthed event. The ground-truth for events such as touch-up, goal, etc., is easy to create and often objective (e.g., Chang et al. [2002], Xiong et al. [2003a], Ariki et al. [2003], Chen et al. [2004], Shih and Huang [2004], and Wang et al. [2004]), whilst the ground-truth tends to be slightly subjective for generic highlight segments (e.g., Rui et al. [2000], Tjondronegoro et al. [2004a, 2004b]). Instead of constructing the ground-truth themselves, some works utilize the skim manually created by independent users to achieve a more objective evaluation. For example, He et al. [1999] have the lecture videos summarized by their authors; Miura et al. [2003] use a review sequence generated by the producer that accompanies a cooking program; Babaguchi et al. [2000] use a summary sequence available on the Internet; and Takahashi et al. [2005] use highlights broadcasted by TV stations. Except for He et al. [1999], all techniques demonstrate a significant degree of overlapping between generated skims/highlights and the ground-truth.

5.3 User Studies

These studies involve independent users judging the quality of generated video abstracts, and are probably the most useful and realistic form of evaluation (especially when keyframes are extracted for user-based interactive tasks such as content browsing and navigation), yet not widely employed due to the difficulty in setting them up. User studies are employed for evaluating keyframe extraction techniques in Drew and Au [2000], Dufaux [2000], and Liu et al. [2003]. Drew and Au [2000] compare their method against Ferman and Tekalp [1998] by counting the number of keyframes that are correct, redundant, and missed. In Dufaux [2000], each keyframe is classified as "good," "fair," or "poor," and their method is demonstrated to be more effective than the selection of middle frames as keyframes. A similar but more complete subjective user study is carried out in Liu et al. [2003] as follows. For each shot, several testers assign a satisfactory score ("good," "acceptable," "bad") regarding its extracted keyframe set (not for each single keyframe, as in Dufaux [2000]). These scores are then used to evaluate the performance of the proposed technique on a large collection of sports, news, home movie, and entertainment videos.

Similarly, for video skimming works, users are asked to directly rate the quality and their satisfaction (i.e., *enjoyability*, *informativeness*) with summaries or how well the summaries have helped the user in interactive tasks such as browsing, searching, and content identification [Lienhart 1997; Christel et al. 1998; Sundaram and Chang 2001; Ma et al. 2002; Li et al. 2003; Ngo et al. 2003; Agnihotri et al. 2004]. For works in which the length of skim is set *a priori*, user satisfaction with different skimming rates can also be compared, for example, Ngo et al. [2003]. The quality of a video skim can also be justified based on user performance with content identification tasks [Smith and Kanade 1998; Erol et al. 2003; Yu et al. 2003; Lu et al. 2004b]. Most techniques demonstrate satisfactory results regarding user acceptance, yet still leave much to be desired, especially regarding the coherence and smoothness of produced skims. Li et al. [2003] and Zhao et al. [2003] report very good results for the abstraction of movies and home videos, respectively.

5.4 Recommendations

In light of a lack of systematic means to evaluate video abstraction techniques, we urge the field to follow the success of NIST TRECVID in furthering the research in shot and scene boundary detection,

high-level feature extraction and video retrieval. Over the last five years, this evaluation benchmark has significantly matured, continuingly proving to be a valuable resource to the video retrieval community. Here, we outline some suggestions towards an open and systematic evaluation framework in a similar manner to TRECVID.

5.4.1 *A Common Dataset.* It is important that video abstraction techniques are evaluated on a common or known dataset, such as TRECVID and Open-Video Archive. This dataset should be large, diverse, and contain full-length video sequences. It should be organized into a genre hierarchy, therefore facilitating the evaluation of domain-specific techniques. It is easier for peers to comprehend the description of a generated abstract if they can watch the original sequence itself. Moreover, authors can make their results available as an abstraction mark-up file and a rendering engine can create the final abstract to be viewed by the peer. It is also easier to organize extensive user studies if everyone can openly access the data.

5.4.2 *Ground-Truthing.* Having a common dataset alone is not sufficient if an effective comparative performance evaluation is to be achieved. Whenever possible, a ground-truth of what the abstract of a specific video sequence should be, in both static and dynamic form, should be established. We strongly believe that a *direct* ground-truth which specifies which video segments or frames are to be part of the video abstract is not applicable to the task, since the same semantic can be conveyed by two different frames or segments. In evaluating a video abstract, the *indirect* ground-truth is both much more useful and easier to establish. Instead of specifying time instances, an indirect ground-truth specifies the semantic content of the abstract, including objects, actions, specific events and the overall story, together with their relative importance. Thus, the ground-truth is itself at a higher semantic level, and will allow multiple solutions. The ground-truth should also be associated with one particular summarization perspective, permitting multiple ground-truths over the same document. Two techniques can only be compared if they agree on the summarization perspective.

5.4.3 *Evaluation Perspectives.* He et al. [1999] list four desirable attributes of a skim for presentation videos that are applicable to any kind of video. These attributes are: (a) *conciseness*, (b) *coverage*, (c) *context*, and (d) *coherence*. The first two attributes are also desirable for a keyframe set. The evaluation of conciseness is straightforward from the length of the produced abstract, while evaluation of coverage can be performed via the abstraction ground-truth, as described before. The evaluation of content and coherence are often subjective and can be carried out by having the user enumerate those segments that are out of context or do not flow well from the previous segment. Other attributes, such as *informativeness* and *interestingness*, are also useful.

We believe, however, that the main focus of the evaluation process should be application-dependent:

—*Browsing*. When video abstracts are used to help the user browse and search for a particular video in a large database, the amount of time spent by the user to complete the task is the most important factor.

—*Computational reduction and content analysis*. When a video abstract is used to reduce the computational complexity in semantic extraction (or video retrieval), an entirely objective evaluation can be achieved by focusing the evaluation on the overall reduction in time and the effectiveness of the produced abstract compared to other representations (including the original sequence) in extracting the target concepts.

—*Story navigation and video editing*. The effectiveness of a keyframe set is measured by how well it helps the user navigate to a desired position on a video sequence. In this case, conciseness is probably less important than coverage.

—*Highlighting*. For video skimming techniques based on the detection of specific highlight events, the evaluation should be carried out on two levels: (1) how well the algorithm detects these events, and (2) the dominance of these events in the abstraction ground-truth. In addition, since generated abstracts are to be watched and enjoyed by the user, coherence is a very important attribute.

The construction of an evaluation benchmark for the video abstraction task is not easy, considering that the TRECVID-2005 team recently had problems producing the ground-truth for the seemingly simple task of camera motion detection. For now, there should be more collaboration among different authors and research groups regarding the sharing of algorithm implementation, data, results, and evaluation tools, encouraging comparative evaluations of different techniques. Authors should at least reach a consensus on what are the best procedures and metrics for evaluating video abstracts, drawing experience from exaisting works, TRECVID, and related fields such as text summarization. Detailed research that focuses exclusively on the evaluation of existing techniques would also be a valuable addition to the field. Our classification scheme already provides a very good reference for selecting compatible techniques for comparison. Even without a ground-truth, we can effectively compare two techniques by putting side-by-side and in random order the video abstracts generated by each technique, and having the user decide which is more representative (or more interesting, informative, etc.) or declaring a "tie" when the user is unsure.

6. CONCLUSIONS

Video abstraction is an integral part of many video applications, including video indexing, browsing, and retrieval. Concisely and intelligently generated video abstracts will facilitate user access to large volumes of video content in an effective and efficient manner. Recently, video abstraction has attracted considerable interest from researchers and as a result, various algorithms and techniques have been proposed. In this work, we have carried out a comprehensive survey and review of the research in two dominant forms of video abstraction: the keyframe set and the skim. We identified important elements and described how they are addressed in specific works. For the extraction of keyframes, these elements are the size of keyframe set, representation scope, base unit, and underlying computational mechanisms, with each element being further categorized. Similarly, for the generation of video skims, five important elements are identified and described, including the overall process, the length of video skim, working data domain, underlying computation mechanism and features used.

Our systematic review of existing works also reveal gaps and opportunities in the field. First of all, although we have outlined some ideas for effective evaluation of video abstraction techniques, there remains a need an extensive study of the subject. Only with a valid evaluation method can the field advance and the best technique for abstracting a video sequence be identified. More emphasis also needs to be put on producing keyframe sets that not only have large content coverage, but are also visually pleasant to the viewer. It is clear that the way a keyframe set is presented to users influences the time and effort they need to grasp the content. However, what the optimal visualization and representation of a given keyframe set should be remains an open question. As for video skimming, more work should be done on maintaining the context and coherence of the generated skim, as the practical deployment of a technique will be limited unless it can generate smoother and more coherent video abstracts. There should be more investigation on the abstraction of raw data such as surveillance and traffic videos, for which the main purpose is to provide the user with an overview of its content or with backward tracking of certain subjects or events. To describe an event in a concise sequence, cinematic principles such as time compression can also be utilized. Finally, to fully generate a highlights program, the video abstract should also be generated according to a narration line, similar to the way that sports highlights in a news program are synchronized to the news reader's speech.

REFERENCES

AGNIHOTRI, L., DIMITROVA, N., AND KENDER, J. R. 2004. Design and evaluation of a music video summarization system. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

AGNIHOTRI, L., KENDER, J., DIMITROVA, N., AND ZIMMERMAN, J. 2005. A framework for personalized multimedia summarization. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR)*.

AIZAWA, K., ISHIJIMA, K., AND SHIINA, M. 2001. Summarizing wearable video. In *Proceedings of the Conference ICIP*. 398–401.

ANER, A. AND KENDER, J. R. 2002. Video summaries through mosaic-based shot and scene clustering. In *Proceedings of the European Conference on Computer Vision* (Denmark).

AOYAGI, S., KOURAI, K., SATO, K., TAKADA, T., SUGAWARA, T., AND ONAI, R. 2004. Implementation of flexible-playtime video skimming. In *Proceedings of the SPIE Multimedia Computin and Networking Conference* vol. 5305. 178–186.

ARIKI, Y., KUMANO, M., AND TSUKADA, K. 2003. Highlight scene extraction in real time from baseball live video. In *Proceedings of the 5th International Workshop on Multimedia Information Retrieval (ACM SIGMM)*. 209–214.

ASSFALG, J., BERTINI, M., COLOMBO, C., BIMBO, A. D., AND NUNZIATI, W. 2003. Semantic annotation of soccer videos: Automatic highlights identification. *Comput. Vision Image Understanding 92*, 2–3, 285–305.

BABAGUCHI, N. 2000. Towards abstracting sports video by highlights. In *Proceedings of the ICME Conference*. New York.

BABAGUCHI, N., KAWAI, Y., AND KITAHASHI, T. 2001. Generation of personalized abstract of sports video. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (Tokyo).

BAGGA, A., HU, J., ZHONG, J., AND RAMESH, G. 2002. Multi-Source combined-media video tracking for summarization. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, vol. 2. 818–821.

CAI, R., LU, L., ZHANG, H.-J., AND YANG, S.-Q. 2003. Highlight sound effects detection in audio stream. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (Baltimore, NY).

CALIC, J. AND IZQUIERDO, E. 2002. Efficient key-frame extraction and video analysis. In *Proceedings of the ITCC Conference*. 28–33.

CALIC, J. AND THOMAS, B. T. 2004. Spatial analysis in key-frame extraction using video segmentation. In *Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)* (Lisboa, Portugal).

CHANG, H. S., SULL, S., AND LEE, S. U. 1999. Efficient video indexing scheme for content-based retrieval. *IEEE Trans. Circ. Syst. Video Technol. 9*, 8 (Dec.), 1269–1279.

CHANG, P., HAN, M., AND GONG, Y. 2002. Extract highlights from baseball game video with hidden markov models. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (Rochester, NY).

CHAU, W.-S., AU, O. C., AND CHONG, T.-S. 2004. Key frame selection by macroblock type and motion vector analysis. In *Proceedings of the ICME Conference* (Taipei, Taiwan).

CHEN, S.-C., SHYU, M.-L., CHEN, M., AND ZHANG, C. 2004. A decision tree-based multimodal data mining framework for soccer goal detection. In *Proceedings of the ICME Conference* (Taipei, Taiwan).

CHIU, P., GIRGENSOHN, A., AND LIU, Q. 2004. Stained-glass visualization for highly condensed video summaries. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

CHRISTEL, M. G., HAUPTMANN, A. G., WARMACK, A. S., AND CROSBY, S. A. 1999. Adjustable filmstrips and skims as abstractions for a digital video library. In *Proceedings of the ADL Conference* (Washington, DC) 98.

CHRISTEL, M. G., SMITH, M. A., TAYLOR, C., AND WINKLER, D. B. 1998. Evolving video skims into useful multimedia abstractions. In *Proceedings of the Conference on Computer-Human Interaction CHI* (Los Angeles, CA). 117–122.

COLDEFY, F. AND BOUTHEMY, P. 2004. Unsupervised soccer video abstraction based on pitch, dominant color and camera motion analysis. In *Proceedings of the ACM Multimedia Conference (ACMMM)*. ACM Press, New York, 268–271.

COLDEFY, F., BOUTHEMY, P., BETSER, M., AND GRAVIER, G. 2004. Tennis video abstraction from audio and visual cues. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing, MMSP* (Siena, Italy).

COOPER, M. AND FOOTE, J. 2002. Summarizing video using non-negative similarity matrix factorization. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing (MMSP)*. St. Thomas, US Virgin Islands. 25–28.

COOPER, M. AND FOOTE, J. 2005. Discriminative techniques for keyframe selection. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*. 502–505.

DAGTAS, S. AND ABDEL-MOTTALEB, M. 2004. Multimodal detection of hightlights for multimedia content. *Multimedia Syst. 9*, 586–593.

DAVID GIBSON, N. C. AND THOMAS, B. 2002. Visual abstraction of wildlife footage using Gaussian mixture models. In *Proceedings of the 15th International Conference on Vision Interface*.

DE SILVA, G. C., YAMASAKI, T., AND AIZAWA, K. 2005. Evaluation of video summarization for a large number of cameras in ubiquitous home. In *Proceedings of the ACM Multimedia Conference (ACMMM)*.

DeMenthon, D., Kobla, V., and Doerman, D. 1998. Video summarization by curve simplification. In *Proceedings of the ACM Multimedia Conference (ACMMM)*. Bristol, UK, 211–218.

Dimitrova, N., McGee, T., and Elenbaas, H. 1997. Video keyframe extraction and filtering: A keyframe is not a keyframe to everyone. In *Proceedings of the 6th International Conference on Information and Knowledge Management*. Las Vegas, Nevada, 113–120.

Divakaran, A., Peker, K. A., Radhakrishnan, R., Xiong, Z., and Cabasson, R. 2003. Video summarization using mpeg-7 motion activity and audio descriptors. In *Video Mining*, A. Rosenfeld et al., Eds. Kluwer Academic.

Divakaran, A., Radhakrishnan, R., and Peker, K. A. 2002. Motion activity-based extraction of key-frames from video shots. In *Proceedings of the IEEE International Conference on Image Processing*. vol. 1. Rochester, NY, 932–935.

Doulamis, A., Doulamis, N., Avrithis, Y., and Kollias, S. 2000. A fuzzy video content representation for video summarization and content-based retrieval. *Signal Processing 80*, 1049–1067.

Doulamis, N. D., Doulamis, A. D., Avrithis, Y. S., and Kollias, S. D. 1998. Video content representation using optimal extraction of Frames and Scenes. In *Proceedings of the ICIP Conference*, vol. 1. Chicago, IL. vol. 1, 875–879.

Doulamis, N. D., Doulamis, A. D., Avrithis, Y. S., and Kollias, S. D. 1999. A stochastic framework for optimal key frame extraction from MPEG vide databases. In *Proceedings of the 3rd Workshop on Multimedia Signal Processing*.

Drew, M. S. and Au, J. 2000. Video keyframe production by efficient clustering of compressed chromaticity signatures. In *Proceedings of the ACM Multimedia Conference*. Los Angeles.

Dufaux, F. 2000. Key frame selection to represent a video. In *Proceedings of the ICIP Conference*. vol. 2. 275–278.

Ekin, A. and Tekalp, A. M. 2002. A framework for tracking and analysis of soccer video. In *Proceedings of the Visual Communication and Image Processing 2002*. Proceedings of SPIE, vol. 4671. San Jose, CA, 763–774.

Erol, B., Lee, D.-S., and Hull, J. 2003. Multimodal summarization of meeting recordings. In *Proceedings of the IEEE International Conference on* Multimedia and Expo. Vol. 3. 25–28.

Fauvet, B., Bouthemy, P., Gros, P., and Spindler, F. 2004. A geometrical key-frame selection method exploiting dominant motion estimation in video. In *Proceedings of the CVIR Conference* (Dublin, Ireland).

Ferman, A. and Tekalp, A. M. 1998. Efficient Filtering and Clustering methods for temporal video segmentation and visual summarization. *J. Visu. Commun. Image Represent. 9*, 4 (Dec.), 336–351.

Ferman, A. M. and Tekalp, A. M. 2003. Two-Stage hierarchical video summary extraction using the GoF color descriptor. *IEEE Trans. Multimedia 5*, 2 (Jun.), 244–256.

Girgensohn, A. 2003. A fast layout algorithm for visual video summaries. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, vol. 2. 77–80.

Girgensohn, A. and Boreczky, J. 1999. Time-Constrained keyframe selection technique. In *Proceedings of the IEEE International Conference on Multimedia and Systems*, vol. 1 (Florence, Italy). 756–761.

Girgensohn, A., Boreczky, J., and Wilcox, L. 2001. Keyframe-Based user interfaces for digital video. *IEEE Comput. 34*, 9 (Sept.), 61–67.

Gong, Y. and Liu, X. 2001. Summarizing video by minimizing visual content redundancies. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Tokyo, Japan).

Gong, Y. and Liu, X. 2003. Video summarization and retrieval using singular value decomposition. *ACM Multimedia Syst. J. 9*, 157–168.

Gong, Y.-H. 2003. Summarizing audio-visual contents of a video program. *EURASIP J. Appl. Signal Process.: Special Issue on Unstructured Info. Manage. Multimedia Data Sources*, 2 (Feb.).

Gu, L., Bone, D., and Reynolds, G. 1999. Replay detection in sports video sequences. In *Proceedings of the Eurographics Workshop on Multimedia* (Milan, Italy).

Gunsel, B. and Tekalp, A. M. 1998. Content-Based video abstraction. In *Proceedings of the ICIP Conference*. (Chicago, Illinois).

Hammoud, R. and Mohr, R. 2000. A probabilistic framework of selecting effective key frames for video browsing and indexing. In *Proceedings of the International Workshop on Real-Time Image Sequence Analysis*.

Han, M., Hua, W., Xua, W., and Gong, Y. 2002. An integrated baseball digest system using maximum entropy method. In *Proceedings of the International Conference on Multimedia and Expo ICME* (Lausanne, Switzerland). 347–350.

Han, S.-H. and Kweon, I.-S. 2005. Scalable temporal interest points for abstraction and classification of video events. In *International Conference on Multimedia and Expo.*.

Hanjalic, A. 2003. Generic approach to highlights extraction from a sport video. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. vol. 1. (Barcelona, Spain). 1–4.

Hanjalic, A., Lagendijk, R., and Biemond, J. 1999. Automated high-level movie segmentation for advanced video retrieval systems. *IEEE Trans. Circ. Syst. Video Technolo. 9*, 4 (June.), 580–588.

HANJALIC, A., LAGENDIJK, R. L., AND BIEMOND, J. 1998. A new method for key frame based video content representation. In *Image Databases and Multi-Media Search*, A. Smeulders and R. Jain, Eds. World Scientific, Singapore.

HANJALIC, A. AND ZHANG, H. 1999. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Trans. Circ. Syst. Video Technolo. 9*, 8 (Dec.), 1280–1289.

HE, L., SANOCKI, E., GUPTA, A., AND GRUDIN, J. 1999. Auto-Summarization of audio-video presentations. In *Proceedings of the ACM Multimedia Conference (ACMMM)* (Orlando, FL). 489–498.

IRANI, M., ANANDAN, P., AND HSU, S. 1998. Video indexing based on mosaic representations. In *Proceedings of the IEEE Conference 86*, 905–921.

JAIMES, A., ECHIGO, T., TERAGUCHI, M., AND SATOH, F. 2002. Learning personalized video highlights from detailed MPEG-7 event metadata. In *Proceedings of the ICIP Conference* (Rochester, NY).

JOSHI, A., AUEPHANWIRIYAKUL, S., AND KRISHNAPURAM, R. 1998. On fuzzy clustering and content based access to networked video databases. In *Proceedings of the International Workshop on Research Issues in Data Engineering (RIDA) Conference*. (Orlando, FL). 42–43.

JUNG, B., KWAK, T., SONG, J., AND LEE, Y. 2004. Narrative abstraction model for story oriented video. In *Proceedings of the ACM Multimedia (ACMMM) Conference*. ACM Press, New York.

KANG, E. K., KIM, S. J., AND CHOI, J. S. 1999. Video retrieval based on scene change detection in compressed domain. *IEEE Trans. Consum. Electron. 45*, 3, 932–936.

KANG, H.-B. 2002. Video abstraction techniques for a digital library. In *Distributed Multimedia Databases: Techniques and Applications*. Idea Group, Hershey, PA. 120–132.

KANG, H.-W. AND HUA, X.-S. 2005. To learn representativeness of video frames. In *Proceedings of the ACM Multimedia Conference*.

KIM, C. AND HWANG, J.-N. 2002. Object-Based video abstraction for video surveillance systems. *IEEE Trans. Circ. Syst. Video Technol. 12*, 12, 1128–1138.

KIM, J.-G., CHANG, H. S., KANG, K., KIM, M., KIM, J., AND KIM, H.-M. 2004. Summarization of news video and its description for content-based access. *International J. Imaging Syst. Technol. 13*, 5, 267–274.

KOBLA, V., DEMENTHON, D., AND DOERMANN, D. 1999. Detection of slow-motion replay sequences for identifying sports videos. In *Proceedings of the IEEE 1999 International Workshop on Multimedia Signal Processing* (Copenhagen, Denmark).

KOMLODI, A. AND MARCHIONINI, G. 1998. Key frame preview techniques for video browsing. In *DL: Proceedings of the 3rd ACM Conference on Digital Libraries*. ACM Press, New York. 118–125.

KOPF, S., HAENSELMANN, T., FARIN, D., AND EFFELSBERG, W. 2004. Automatic generation of video summaries for historical films. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

KRAAIJ, W., SMEATON, A., P.OVER, AND ARLANDIS, J. 2004. Trecvid 2004 - An overview. In *Proceedings of the NIST TRECVID Conference*.

LATECKI, L. J., WIDLDT, D. D., AND HU, J. 2001. Extraction of key frames from videos by optimal color composition matching and polygon simplification. In *Proceedings of the Multimedia Signal Processing Conference* (Cannes, France).

LEE, H. AND KIM, S. 2002. Rate-driven key frame selection using temporal variation of visual content. *Electroni. Lette. 38*, 5 (Feb.), 217–218.

LEE, H.-C. AND KIM, S.-D. 2003. Iterative key frame selection in the rate-constraint environment. *Signal Process.: Image Commun.* 18, 1–15.

LEE, J., OH, J., AND HWANG, S. 2005. Scenario based dynamic video abstractions using graph matching. In *ACM Multimedia Conference (ACMMM)*. 810–809.

LEE, M.-C., CHEN, W.-G., LIN, C. C. G., MARKOC, T., ZABINSKY, S., AND SZELISKI, R. 1997. A layered video object coding system using sprite and affine motion model. *IEEE Trans. Circ. Syst. Technol. 7*, 1, 130–145.

LEE, S. AND HAYE, M. 2004. An application for interactive video abstraction. In *Proceedings of the ICASSP Conference*.

LEE, S.-H., YEH, C. H., AND KUO, C. C. J. 2004. Video skimming based on story units via general tempo analysis. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*(Taipei, Taiwan).

LEINHART, R. W. AND YEO, B.-L. 2003. Method and apparatus for abstracting video data. US Patent 6597859 by Intel Corporation.

LI, Y., NARAYANAN, S., AND KUO, C.-C. J. 2003. Movie content analysis, indexing and skimming via multimodal information. In *Video Mining*, A. Rosenfeld et al., Eds. Kluwer Academic Hingham, MA.

LI, Y., ZHANG, T., AND TRETTER, D. 2001. An overview of video abstraction techniques. Tech. Rep. HP-2001-191, HP Laboratory. July.

LI, Z., SCHUSTER, G., KATSAGGELOS, A. K., AND GANDHI, B. 2004. Optimal video summarization with a bit budget constraint. In *Proceedings of the IEEE International Conference on Image Processing* (Singapore).

LIE, W.-N. AND LAI, C.-M. 2004. News video summarization based on spatial and motion feature analysis. In *Proceedings of the 5th Pacific Rim Conference on Multimedia*. Lecture Notes in Computer Science, vol. 3332. 246 – 255.

LIENHART, R. 1997. Dynamic summarization of home video. In *Proceedings of the SPIE Conference on Storage and Retrieval for Media Databases*. vol. 3972. 378–389.

LIENHART, R., PFEIFFER, S., AND EFFELSBERG, W. 1997. Video abstracting. *Commun. ACM 40*, 12 (Dec.), 54–63.

LIU, T. AND KENDER, J. R. 2002a. An efficient error-minimizing algorithm for variable-rate temporal video sampling. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*.

LIU, T. AND KENDER, J. R. 2002b. Optimization algorithms for the selection of key frame sequences of variable length. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

LIU, T. AND KENDER, J. R. 2002c. Rule-Based semantic summarization of instructional videos. In *Proceedings of the International Conference on Image Processing (ICIP)*.

LIU, T., ZHANG, H.-J., AND QI, F. 2003. A novel video key-frame extraction algorithm based on perceived motion energy model. *IEEE Trans. Circ. Syst. Video Technol. 13*, 10 (Oct.), 1006–1013.

LIU, T., ZHANG, X., FENG, J., AND LO, K. 2004. Shot reconstruction degree: A novel criterion for keyframe selection. *Pattern Recogn. Lett. 25*, 12 (Sept.), 1451–1457.

LU, S., KING, I., AND LYU, M. 2003. Video summarization using greedy method in a constraint satisfaction framework. In *Proceedings of the 9th International Conference on Distributed Multimedia Systems (DMS)* (Miami, FL), 456–461.

LU, S., KING, I., AND LYU, M. 2004a. Video summarization by video structure analysis and graph optimization. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

LU, S., LYU, M., AND KING, I. 2004b. Video summarization by spatial-temporal graph optimization. In *Proceedings of the IEEE Symposium on Circuits and Systems (ISCAS)* (Vancouver, Canada).

LU, S., LYU, M., AND KING, I. 2005. Semantic video summarization using mutual reinforcement principle and shot arrangement patterns. In *Proceedings of the 11th International Multimedia Modeling Conference (MMM)* (Melbourne, Australia).

MA, Y.-F., LU, L., ZHANG, H.-J., AND LI, M. 2002. A user attention model for video summarization. In *Proceedings of the ACM Multimedia Conference (ACMMM)* (Juan Les Pin, France).

MARLOW, S., SADLIER, D., O'CONNOR, N., AND MURPHY, N. 2002. Audio processing for automatic TV sports program highlights. In *Proceedings of the Irish Signals and Systems Conference*.

MASUMITSU, K. AND ECHIGO, T. 2000. Video summarization using reinforcement learning in eigenspace. In *Proceedings of the Conference ICIP*, vol. 2. New York 267–270.

MAULDIN, M. L., SMITH, M. A., STEVENS, S. M., WACTLAR, H. D., CHRISTEL, M. G., AND REDDY, D. R. 1997. US5664227: System and method for skimming digital audio/video data. US Patent US5664227 by Carnegie Mellon University, Pittsburgh, PA.

MEI, T., ZHU, C., ZHOU, H., AND HUAZ, X. 2005. SpatioTemporal quality assessment for home videos. In *Proceedings of the ACM Multimedia Conference (ACMMM)*.

MIURA, K., HAMADA, R., IDE, I., SAKAI, S., AND TANAKA, H. 2003. Motion based automatic abstraction of cooking videos. In *IPSJ Trans. Comput. Vis. Image Media 44*.

MIYAUCHI, S., BABAGUCHI, N., AND KITAHASHI, T. 2003. Highlight detection and indexing in broadcast sports video by collaborative processing of text, audio, and image. *Syst. Comput. Japan 34*, 12, 22 – 31.

NARASIMHA, R., SAVAKIS, A., RAO, R., AND QUEIROZ, R. D. 2003. Key frame extraction using MPEG-7 motion descriptors. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers* (Pacific Grove, CA).

NG, H. W., SAWAHATA, Y., AND AIZAWA, K. 2002. Summarization of wearable videos using support vector machine. In *Proceedings of the International Conference on Multimedia and Expo ICME*. vol. 1. 325–328.

NGO, C.-W., MA, Y.-F., AND ZHANG, H.-J. 2003. Automatic video summarization by graph modeling. In *Proceedings of the International Conference on Computer Vision (ICCV)*. vol. 1 (Nice, France).

OMOIGUI, N., HE, L., GUPTA, A., GRUDIN, J., AND SANOCKI, E. 1999. Time-compression: Systems concerns, usage, and benefits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 136–143.

OTSUKA, I., NAKANE, K., DIVAKARAN, A., HATANAKA, K., AND OGAWA, M. 2005. A highlight scene detection and video summarization system using audio feature for a personal video recorder. *IEEE Trans. Consumer Electron. 51*, 112–116.

OUYANG, J.-Q., LI, J.-T., AND ZHANG, Y.-D. 2003. Replay boundary detection in MPEG compressed video. In *Proceedings of the Machine Learning and Cybernetics International Conference*, vol. 5.

PAN, H., BEEK, P., AND SEZAN, M. 2001. Detection of slow-motion replay segments in sports video for highlights generation. In *Proceedings of the ICASSP Conference* (Salt Lake City, UT).

PARSHIN AND CHEN, L. 2000. Implementation and analysis of several keyframe-based browsing interfaces to digital video. Lecture Notes on Computer Science, vol. 1923, 206.

PARSHIN AND CHEN, L. 2004. Video summarization based on user-defined constraints and preferences. In *Proceedings of the RIAO Conference*.

PEKER, K., DIVAKARAN, A., AND SUN, H. 2001. Constant pace skimming and temporal sub-sampling of video using motion activity. In *Proceedings of the ICIP Conference*. Thessaloniki, Greece, 414–417.

PEKER, K. A. AND DIVAKARAN, A. 2004. Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure. In *Proceedings of the ICME Conference* (Taipei, Taiwan).

PETKOVIC, M., MIHAJLOVIC, V., AND JONKER, W. 2002. Multi-modal extraction of highlights from TV formula 1 programs. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (Lausanne, Switzerland).

PEYRARD, N. AND BOUTHEMY, P. 2003. Motion based selection of relevant video segments for video summarization. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (Baltimore, MD).

PFEIFFER, S., LIENHART, R., FISCHER, S., AND EFFELSBERG, W. 1996. Abstracting digital movies automatically. *J. Vis. Commun. Image Represent. 7*, 4 (Dec.), 345–353.

POPE, A., KUMAR, R., SAWHNEY, H., AND WAN, C. 1998. Video abstraction: Summarizing video content for retrieval and visualization. In *Proceedings of the 32nd Asilomar Conference on Signals, Systems and Computers*. 915–919.

PORTER, S. V., MIRMEHDI, M., AND THOMAS, B. T. 2003. A shortest path representation for video summarisation. In *Proceedings of the 12th International Conference on Image Analysis and Processing*. 460–465.

RADHAKRISHNAN, R., DIVAKARAN, A., AND XIONG, Z. 2004. A time series clustering based framework for multimedia mining and summarization using audio features. In *MIR: Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*. ACM Press, New York, 157–164.

RADHAKRISHNAN, R., XIONG, Z., DIVAKARAN, A., AND KAN, T. 2004. Time series analysis and segmentation using eigenvectors for mining semantic audio label sequences. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*.

RASHEED, Z. AND SHAH, M. 2003. Scene detection in hollywood movies and TV shows. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference* (Madison, WI).

RONG, J., JIN, W., AND WU, L. 2004. Key frame extraction using inter-shot information. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

RUI, Y., GUPTA, A., AND ACERO, A. 2000. Automatically extracting highlights for TV baseball programs. In *Proceedings of the ACM Multimedia Conference (ACMMM)* (Los Angeles, CA).

RUSSELL, D. M. 2000. A design pattern-based video summarization technique: Moving from low-level signals to high-level structure. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)* (Washington, DC). 3048.

SHIH, H.-C. AND HUANG, C.-L. 2004. Detection of the highlights in baseball video program. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

SMITH, M. A. AND KANADE, T. 1998. Video skimming and characterization through the combination of image and language understanding. In *Proceedings of the International Workshop on Content-Based Access of Image and Video Databases (CAIVD)* (Bombay, India), 61–70.

SUGANO, M., NAKAJIMA, Y., YANAGIHARA, H., AND YONEYAMA, A. 2004. Generic summarization technology for consumer video. In *Proceedings of the 5th Pacific Rim Conference on Multimedia (PCM)*. Lecture Notes in Computer Science, vol. 3332.

SULL, S., KIM, J.-R., KIM, Y., CHANG, H. S., AND LEE, S. U. 2001. Scalable hierarchical video summary and search. In *Proceedings of the SPIE Conference on Storage and Retrieval for Media Databases*, vol. 4315, 553–561.

SUN, X. AND KANKANHALLI, M. S. 2000. Video summarization using r-sequences. *J. Real Time Imaging 6*, 449–459.

SUNDARAM, H. AND CHANG, S.-F. 2001. Condensing computable scenes using visual complexity and film syntax analysis. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* (Tokyo, Japan).

SUNDARAM, H. AND CHANG, S.-F. 2002. Video skims: Taxonomies and an optimal generation framework. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (Rochester, NY).

TAKAHASHI, Y., NITTA, N., AND BABAGUCHI, N. 2005. Video summarization for large sports video archives. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*. 1170–1173.

TANIGUCHI, Y., AKUTSU, A., AND TONOMURA, Y. 1997. PanoramaExcerpts: Extracting and packing panoramas for video browsing. In *Proceedings of the ACM Multimedia Conference (ACMMM)*. ACM Press, New York, 427–436.

TASKIRAN, C. M., AMIR, A., PONCELEON, D. B., AND DELP, E. J. 2001. Automated video summarization using speech transcripts. In *Proceedings of the SPIE Conference on Storage and Retrieval for Media Databases*, vol. 4676. 371–382.

TEODOSIO, L. AND BENDER, W. 1993. Salient video stills: Content and context preserved. In *Proceedings of the ACM Multimedia Conference (ACMMM)*. ACM Press, New York, 39–46.

TIE-YAN LIU, X.-D. Z. 2000. Inertia-based cut detection technique: a step to the integration of video coding and content-based retrieval. *Proceedings of the International Conference on Signal Processing (ICSP) 2*, 1018–1025.

TJONDRONEGORO, D., CHEN, Y.-P. P., AND PHAM, B. 2004a. Integrating highlights for more complete sports video summarization. *Proceedings of the IEEE MultiMedia 11*, 4, 22–37.

TJONDRONEGORO, D. W., CHEN, Y.-P. P., AND PHAM, B. 2004b. Classification of self-consummable highlights for soccer video summaries. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

TONOMURA, Y., AKUTSU, A., OTSUJI, K., AND SADAKATA, T. 1993. VideoMap and video SpaceIcon: Tools for anatomizing video content. In *Proceedings of the INTERCHI Conference*. 131–136.

TRUONG, B. T., VENKATESH, S., AND DORAI, C. 2004. Discovering semantics from the visualization of film takes. In *Proceedings of the IEEE Multimedia Modeling (MMM)* (Brisbane, Australia). 109–116.

UCHIASHI, S., FOOTE, J., GIRGENSOHN, A., AND BORECZKY, J. 1999. Video manga: Generating semantically meaningful video summaries. In *Proceedings of the ACM Multimedia Conference (ACMMM)* (Orlando, FL), 383–392.

VERMAAK, J., PÉREZ, P., GANGNET, M., AND BLAKE, A. 2002. Rapid summarisation and browsing of video sequences. In *Proceedings of the British Machine Vision Conference, (BMVC)*, vol. 1. (Cardiff, UK).

WAN, K. AND XU, C. 2004a. Efficient multimodal features for automatic soccer highlight generation. In *Proceedings of the ICPR Conference*, vol. 3. 973–976.

WAN, K. AND XU, C. 2004b. Robust soccer highlight generation with a novel dominant-speech feature extractor. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

WAN, K., YAN, X., AND XU, C. 2005. Automatic mobile sports highlights. In *Proceedings of the ICME Conference*. 638–641.

WANG, J., XU, C., E.S.CHNG, AND TIAN, Q. 2004. Sports highlight detection from keyword sequences using HMM. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

WANG, P., CAI, R., AND YANG, S.-Q. 2004. Contextual browsing for highlights in sports video. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

WOLF, W. 1996. Keyframe selection by motion analysis. In *Proceedings of the ICASSP Conference*, vol. 2. 1228–1231.

WU, J. K., KANKANHALLI, M. S., LIM, J.-H., AND HONG, D. 2000. *Perspective on Content-Based Multimedia Systems*. Kluwer Academic, Hingham, MA.

XIE, Y.-X., LUAN, X.-D., LAO, S.-Y., WU, L.-D., XIAO, P., AND WEN, J. 2004. EDU: A model of video summarization. In *Proceedings of the CIVR Conference*. Lecture Notes in Computer Science, vol. 3115 (Dublin, Ireland). 106–114.

XIONG, W., LEE, J. C.-M., AND MA, R.-H. 1997. Automatic video data structuring through shot partitioning and key frame computing. *Mach. Vis. Appl. 10*, 2, 51–65.

XIONG, Z., RADHAKRISHNAN, R., AND DIVAKARAN, A. 2003a. Generation of sports highlights using motion activity in combination with a common audio feature extraction framework. In *Proceedings of the ICIP Conference*, vol. 1 (Barcelona, Spain). 5–8.

XIONG, Z., RADHAKRISHNAN, R., DIVAKARAN, A., AND HUANG, T. S. 2003a. Audio-based highlights extraction from baseball, golf and soccer games in a unified framework. In *Proceedings of the ICASSP Conference* (Hong Kong, China).

XIONG, Z., RADHAKRISHNAN, R., DIVAKARAN, A., AND HUANG, T. S. 2004. Effective and efficient sports highlight extraction using the minimum description length criterion in selecting GMM structures. In *Proceedings of the International Conference on Multimedia and Expo (ICME)* (Taipei, Taiwan).

YAHIAOUI, I., MERIALDO, B., AND HUET, B. 2001. Automatic video summarization. In *Proceedings of the CBMIR Conference*.

YEUNG, M. M. AND LEO, B.-L. 1995. Efficient matching and clustering of shots. In *Proceedings of the Conference ICIP. 2*, 338–341.

YEUNG, M. M. AND LEO, B.-L. 1997. Video visualization for compact representation and fast browsing of pictorial content. *IEEE Trans. Circ. Syst. Video Technol. 7*, 5 (Oct.).

YU, B., MA, W.-Y., NAHRSTEDT, K., AND ZHANG, H.-J. 2003. Video summarization based on user log enhanced link analysis. In *Proceedings of the ACM Multimedia Conference (ACMMM)* (Berkeley, CA).

YU, X.-D., WANG, L., TIAN, Q., AND XUE, P. 2004. Multi-level video representation with application to keyframe extraction. In *Proceedings of the International Conference on Multimedia Modeling (MMM)*. 117–121.

ZHANG, H., WU, J., ZHONG, D., AND SMOLIAR, S. 1997. An integrated system for content-based video retrieval and browsing. *Pattern Recogn. 30*, 4, 643–658.

ZHANG, X.-D., LIU, T.-Y., LO, K.-T., AND FENG, J. 2003. Dynamic selection and effective compression of key frames for video abstraction. *Pattern Recogn. Lett. 24*, 9-10, 1523–1532.

ZHAO, L., QI, W., LI, S., S.Q.YANG, AND ZHANG, H. 2000. A new content-based shot retrieval approach: Key-frame extraction based nearest feature line (NFL) classification. In *Proceedings of the ACM Multimedia Information Retrieval Conference* (Los Angeles, CA). 217–220.

ZHAO, M., BU, J., AND CHEN, C. 2003. Audio and video combined for home video abstraction. In *Proceedings of the International Conference on Acoustic, Speech and Signal Processing*, vol. 5 (Hong Kong, Japan). 620–623.

ZHUANG, Y., RUI, Y., HUANG, T., AND MEHROTRA, S. 1998. Adaptive key frame extraction using unsupervised clustering. In *Proceedings of the International Conference on Image Processing (ICIP)* (Chicago, IL). 866–870.

Received October 2004; revised February 2006; accepted February 2006