J Real-Time Image Proc (2006) 1:69–88
DOI 10.1007/s11554-006-0001-1

ORIGINAL RESEARCH PAPER

An innovative algorithm for key frame extraction in video summarization

Ciocca Gianluigi · Schettini Raimondo

Received: 22 February 2006 / Accepted: 9 May 2006 / Published online: 14 July 2006

Abstract Video summarization, aimed at reducing the amount of data that must be examined in order to retrieve the information desired from information in a video, is an essential task in video analysis and indexing applications. We propose an innovative approach for the selection of representative (key) frames of a video sequence for video summarization. By analyzing the differences between two consecutive frames of a video sequence, the algorithm determines the complexity of the sequence in terms of changes in the visual content expressed by different frame descriptors. The algorithm, which escapes the complexity of existing methods based, for example, on clustering or optimization strategies, dynamically and rapidly selects a variable number of key frames within each sequence. The key frames are extracted by detecting curvature points within the curve of the cumulative frame differences. Another advantage is that it can extract the key frames on the fly: curvature points can be determined while computing the frame differences and the key frames can be extracted as soon as a second high curvature point has been detected. We compare the performance of this algorithm with that of other key frame extraction algorithms based on different approaches. The summaries obtained have been objectively evaluated by three quality measures: the Fidelity measure, the Shot Reconstruction Degree measure and the Compression Ratio measure.

Keywords Video summarization · Visual summary evaluation · Dynamic key frames extraction · Frame content description

1 Introduction

The growing interest of consumers in the acquisition of and access to visual information has created a demand for new technologies to represent, model, index and retrieve multimedia data. Very large databases of images and videos require efficient algorithms that enable fast browsing and access to the information pursued [1]. In the case of videos, in particular, much of the visual data offered are simply redundant, and we must find a way to retain only the information strictly needed for functional browsing and querying.

Video summarization, aimed at reducing the amount of data that must be examined in order to retrieve a particular piece of information in a video, is consequently an essential task in applications of video analysis and indexing [2], as can be seen in Fig. 1. Generally, a video summary is a sequence of still or moving images, with or without audio. These images must preserve the overall contents of the video with minimum data. Still images chronologically arranged form a pictorial summary that can be assumed to be the equivalent of a video storyboard. Summarization utilizing moving images (and at times a corresponding audio abstract) is called video skimming; the product is similar to a video trailer or clip. Both approaches must

C. Gianluigi (✉) · S. Raimondo
Dipartimento di Informatica Sistemistica e Comunicazione
(DISCo), Università degli studi di Milano-Bicocca,
Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy
e-mail: ciocca@disco.unimib.it

S. Raimondo
e-mail: schettini@disco.unimib.it

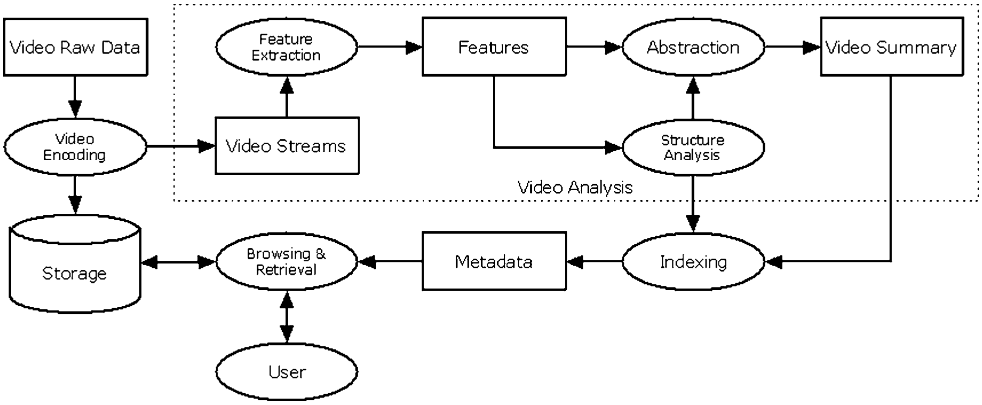


Fig. 1 General application of the video analysis and indexing tasks

present a summary of the important events recorded in the video. We focus our attention here on the creation of a visual summary using still images, called key frames, extracted from the video. Although video skimming conveys pictorial, motion and, where used, audio information, still images can summarize the video content in more rapid and compact way: users can grasp the overall content more quickly from key frames than by watching a set of video sequences (even when brief). Key frame-based video representation views video abstraction as a problem of mapping an entire segment (both static and moving contents) to some small number of representative images. The extraction of key frames must be automatic and content based so that they maintain the salient content of the video while avoiding all redundancy. In theory, semantic primitives of the video such as relevant objects (people or object identities), actions and events (a meeting, a convention, etc.) should be used.

However, since this kind of specific semantic analysis is not currently feasible, key frame extraction based on low-level video features (mainly visual features) is used instead [1]. Besides providing video browsing capability and content description, key frames act as video "bookmarks" that designate interesting events captured, supplying direct access to video sub-sequences. Key frames, which visually represent the video content, can also be used in the indexing process, where we can apply the same indexing and retrieval strategies developed for image retrieval to retrieve video sequences. Low-level visual features can be used in indexing the key frames and thus the video sequences to which they belong. The use of low-level features in the indexing process should not be considered as a less powerful strategy; high-level features with different levels of semantic can be derived from them. For example, high-level information can be derived from low-level visual features by using knowledge-based techniques inherited from the domain of artificial intelligence and pattern recognition. Examples of this can be found in Antani et al. [3], where different pattern recognition strategies are discussed, and in Schettini et al. [4], where classification strategies are used to annotate the global content of the images in terms of high-level concepts (close-up, indoor, outdoor). A similar strategy can be adopted to automatically identify semantic regions within the images themselves [5]. If the audio track is available, transcripts of the audio can also be used [6].

Section 2 of this paper presents several approaches to key frame extraction described in the literature. Section 3 introduces the three quality measures used here to evaluate the summaries. Section 4 describes the key frame extraction algorithm proposed. The results of the comparison of our approach with other key frame selection algorithms are shown in Sects. 5 and 6. Section 7 briefly illustrates an application of our algorithm.

2 Related work

Different methods can be used to select key frames. In general these methods assume that the video has already been segmented into shots (a continuous sequences of frames taken over a short period of time) by a shot detection algorithm and extract the key frames from within each shot. One of the possible approaches to key frame selection is to take the first frame in the shot as the key frame [7]. Ueda et al. [8] and Rui et al. [9] use the first and last frames of each shot. Other approaches time sample the shots at predefined intervals, as in Pentland et al. [10] where the key frames are taken from a set location within the shot or in an alternative approach where the video is time sampled regardless of shot boundaries [9]. These approaches do not consider the dynamics in the visual content of the shot but rely on the information regarding the sequence's boundaries. They often extract a fixed number of key frames per shot. In Zhonghua et al. [11] only one key frame is extracted from each shot: the frames are segmented into objects and background, and the frame with the maximum ratio of objects to background is chosen as the key frame of the segment since it is assumed to convey the most information about the shot. Other approaches try to group the key frames (in each shot or in the whole video) into visually similar clusters. In Arman et al. [12] the shot is compacted into a small number of frames, grouping consecutive frames together. Zhuang et al. [13] group the frames in clusters, and the key frames are selected from the largest clusters. In Girgensohn et al. [14] constraints on the position of the key frames in time are also used in the clustering process; a hierarchical clustering reduction is performed, obtaining summaries at different levels of abstraction. In Gong and Liu [15] the video is summarized with a clustering algorithm based on single value decomposition (SVD). The video frames are time sampled and visual features computed from them. The refined feature space obtained by the SVD is clustered, and a key frame is extracted from each cluster.

In order to take into account the visual dynamics of the frames within a sequence, some approaches compute the differences between pairs of frames (not necessarily consecutive) in terms of color histograms, motion or other visual descriptions. The key frames are selected by analyzing the values obtained. Zhao et al. [16] have developed a simple method for key frame extraction called Simplified Breakpoints. A frame is selected as a key frame if its color histograms differ from that of the previous frame by a given threshold. When the set of selected frames reaches the required number of key frames for the shot, the process stops. In Hanjalic et al. [17] frame differences are taken to build a "content development" curve that is approximated, using an error minimization algorithm, by a curve composed of a predefined number of rectangles. Hoon et al. [18] propose a very simple approach: the key frames are selected by an adaptive temporal sampling algorithm that uniformly samples the y-axis of the curve of cumulative frame differences. The resulting nonuniform sampling on the curve's x-axis represents the set of key frames.

The compressed domain is often considered when developing key frame extraction algorithms since it easily allows to express the dynamics of a video sequence through motion analysis. Narasimha et al. [19] propose a neural network approach using motion intensities computed from MPEG-compressed video. A fuzzy system classifies the motion intensities into five categories, and those frames that exhibit high intensities are chosen as key frames. In Calic and Izquierdo [20] video features extracted from the statistics on the macro-blocks of an MPEG-compressed video are used to compute frame differences. A discrete contour evolution algorithm is applied to extract key frames from the curve of the frame differences. In Liu et al. [21] a perceived motion energy (PME) computed on the motion vectors is used to describe the video content. A triangle model is then employed to model motion patterns and extract key frames at the turning points of acceleration and deceleration.

The drawback of most of these approaches is that the number of representative frames must be set in some manner a priori depending on the length of the video shots for example. This cannot guarantee that the frames selected will not be highly correlated. It is also difficult to set a suitable interval of time, or frames: large intervals mean a large number of frames will be chosen, while small intervals may not capture enough representative frames and those chosen may not be in the right places to capture significant content. Still other approaches work only on compressed video, are threshold-dependent or are computationally intensive (e.g., [21, 22]).

In this paper, we instead propose an approach for the selection of key frames that determines the complexity of the sequence in terms of changes in the pictorial content using three visual features: its color histogram, wavelet statistics and an edge direction histogram. Similarity measures are computed for each descriptor and combined to form a frame difference measure. The frame differences are then used to dynamically and rapidly select a variable number of key frames within each shot. The method woks fast on all kind of videos (compressed or not) and does not exhibit the complexity of existing methods based, for example, on clustering strategies. It can also extract key frames on the fly, i.e., it can output key frames while computing the frame differences without having to process the whole shot.

3 Summary evaluation

One of the most challenging topics in the field of video analysis and summarization is that of evaluating the summaries produced by the different key frame extraction algorithms. In their work to design a framework for video summarization, Fayzullin et al. [23] define three properties that must be taken into account when creating a video summary: continuity, priority and repetition. Continuity means that the summarized video must be as uninterrupted as possible. Priority means that, in a given application, certain objects or events may be more important than others, and thus the summary must contain high-priority items. Repetition means that it is important to not represent the same events over and over again. It is often very difficult to successfully incorporate these semantic properties in a summarization algorithm. Priority, in particular, is a highly task-dependent property. It requires that video experts carefully define the summarization rules most suitable for each genre of video sequence processed.

The most common evaluation of a summary relies on the subjective opinion of a panel of users. This shifts the problem of incorporating semantic information into the summarization algorithm to the evaluation phase where users are requested to compare the summary with the original sequence. For example, in Narasimha et al. [19] and Lagendjik et al. [24] a global subjective evaluation is given for the goodness of the summary; in Liu et al. [21] users are asked to give scores based on their satisfaction marking the summaries as good, acceptable or bad. Ngo et al. [25] apply the criteria of ‘‘informativeness’’ and ‘‘enjoyability’’, for their evaluation of video highlights: ‘‘informativeness’’ assesses the capability of covering the content while avoiding redundancy; ‘‘enjoyability’’ assesses the performance of the algorithm in selecting perceptually agreeable video segments for summaries.

The problem of these approaches is that their evaluation is highly subjective and cannot be used to analyze video sequences automatically. We have chosen, instead, a more objective, general purpose to summary evaluation, one that does not take into account the kind of video being processed, and can be automatically applied to all video sequences without requiring the services of video experts. A summary is considered good if the set of key frames effectively represents the pictorial content of the video sequence. This objective evaluation is valid, regardless of genre, and can be performed automatically if a suitable quality measure is provided. From the very few works that have addressed the problem of objective evaluation of summaries, we have chosen two quality measures. The first, well known in the literature, is the Fidelity measure, as proposed by Chang et al. [26]; the second is the Shot Reconstruction Degree (SRD) recently proposed by Liu et al. [27]. These measures were chosen because they apply two different approaches: the Fidelity employs a global strategy, while the SRD uses a local evaluation of the key frames. Along with the evaluation of the pictorial content using these two measures we have also judged the compactness of the summary on the basis of the Compression Ratio measure.

3.1 Fidelity

The Fidelity measure, which compares each key frame in the summary with the other frames in the video sequence, is defined as a semi-Hausdorff distance. We let a video sequence starting at time t and containing γ_{NF} frames be

$$\mathbf{S}_t = \{F(t+n) | n = 0, 1, \ldots, \gamma_{\mathrm{NF}} - 1\} \tag{1}$$

and the set of γ_{NKF} key frames extracted from the video sequence be

$$\mathbf{KF}_t = \{F_{\mathrm{KF}}(t+n_1), F_{\mathrm{KF}}(t+n_2), \ldots, F_{\mathrm{KF}}(t+n_{\mathrm{NKF}}) | 0 \leq n_i \tag{2}$$

The distance between the set of key frames $\mathbf{KF}_t$ and a frame F belonging to the video sequence $\mathbf{S}_t$ can be computed as

$$d(F(t+n), \mathbf{KF}_t) = \min_j \{\mathrm{Diff}(F(t+n), F_{\mathrm{KF}}(t+n_j))\} \quad j = 1, 2, \ldots, \gamma_{\mathrm{NKF}}, \tag{3}$$

where Diff( ) is a suitable frame difference measure. The distance between the video sequence $\mathbf{S}_t$ and the set of key frames $\mathbf{KF}_t$ is finally defined as

$$d(\mathbf{S}_t, \mathbf{KF}_t) = \max_n \{d(F(t+n), \mathbf{KF}_t) | n = 0, 1, \ldots, \gamma_{\mathrm{NF}} - 1\}. \tag{4}$$

We can then compute the Fidelity measure as

$$\mathrm{Fidelity}(\mathbf{S}_t, \mathbf{KF}_t) = \mathrm{MaxDiff} - d(\mathbf{S}_t, \mathbf{KF}_t), \tag{5}$$

where MaxDiff is the largest possible value that the Diff( ) frame difference distance can assume. High Fidelity values indicate that the key frames extracted from the video sequence provide a good global description of the visual content of the sequence. A visual representation of the Fidelity measure computation is shown in Fig. 2.

3.2 Shot Reconstruction Degree

The idea underlying the SRD measure is that, using a suitable frame interpolation algorithm, we should be able to reconstruct the whole sequence, from the set of key frames. The better the reconstruction approximates the original video sequence, the better the key frames summarize its content. Let the video sequence $\mathbf{S}_t$ and the set of key frames $\mathbf{KF}_t$ be defined as in the previous paragraph. Let FIA( ) be a frame interpolation algorithm that computes a frame from a pair of key frames in $\mathbf{KF}_t$:

$$\tilde{F}(t+n) = \mathrm{FIA}\left(F_{\mathrm{KF}}(t+n_j), F_{\mathrm{KF}}(t+n_{j+1}), n, n_j, n_{j+1}\right) \quad n_j \leq n < n_{j+1}. \tag{6}$$

The SRD measure is defined as

$$\mathrm{SRD}(\mathbf{S}_t, \mathbf{KF}_t) = \sum_{n=0}^{\gamma_{\mathrm{NF}}-1} \mathrm{Sim}\left(F(t+n), \tilde{F}(t+n)\right), \tag{7}$$

where Sim( ) is a similarity measure between two frames. In Liu et al. [27], the chosen similarity measure has been defined as a PSNR-like measure:

$$\mathrm{Sim}\left(F(t+n), \tilde{F}(t+n)\right) = C \log\left(\mathrm{MaxDiff} / \mathrm{Diff}\left(F(t+n), \tilde{F}(t+n)\right)\right), \tag{8}$$

where C is a positive constant, Diff( ) a frame difference distance computed on a gray scale version of the original frames and MaxDiff the largest possible value it can assume. The SRD measure focuses on local details within the video sequences; it should thus be able to evaluate more accurately the video summary with respect to the visual dynamics of the original video. A visual representation of the SRD computation is shown in Fig. 3.

3.3 Compression ratio

A video summary should not contain too many key frames since the aim of the summarization process is to allow users to quickly grasp the content of a video sequence. For this reason, we have also evaluated the compactness of the summary (compression ratio). The compression ratio is computed by dividing the number of key frames in the summary by the length of the video sequence. For a given video sequence $\mathbf{S}_t$, the compression rate is thus defined as

$$\mathrm{CRatio}(\mathbf{S}_t) = 1 - \gamma_{\mathrm{NKF}}/\gamma_{\mathrm{NF}}, \tag{9}$$

where γ_{NKF} is the number of key frames in the summary and γ_{NF} the total number of frames in the video sequence. Ideally, a good summary produced by a key frame extraction algorithm will present both high-quality measure (in terms of Fidelity or SRD) and a high compression ratio (i.e., small number of key frames).

4 New key frame extraction approach

The proposed key frame extraction algorithm functions as shown in Fig. 4. The algorithm employs the information obtained by a video segmentation algorithm to process each shot. We have developed a prototypical segmentation algorithm that for the moment detects only abrupt changes and fades, because these are the most common editing effects. For abrupt changes we have implemented a threshold-based algorithm coupled with a frame difference measure computed from histograms and texture descriptors. The same frame difference measure is used for the key frame extraction algorithm. To detect fades we have implemented a modified version of the algorithm proposed by Fernando et al. [28]. The results obtained by these algorithms are submitted for evaluation to a decision module which gives the final response. This allows us to cope with conflicting results or with groups of frames that are not meaningful, such as those between the end of a fade-out and the start of a fade-in, increasing the robustness of the detection phase. A gradual transition detection algorithm is currently being developed and will be integrated in a similar manner.

We distinguish two types of shots: ‘‘informative’’ shots (type A) and ‘‘uninformative’’ shots (type B). Type B shots are those limited by: a fade-out followed by a fade-in effect, a fade-out followed by a cut or a cut followed by a fade-in. If we were to extract key frames from these shots the resulting set of frames would contain uniformly colored images that are meaningless in terms of the information supplied. Key frames are extracted from type A shots only.

4.1 Frame difference measure

In general, a single visual descriptor cannot capture all the pictorial details needed to estimate the changes in the visual content of frames and the visual complexity of a video shot. In defining what a good pictorial representation of a frame is, we must take into account both color properties and structural properties, such as texture. Instead, as stated in Sect. 2, many existing algorithms use only one feature. To overcome the frame representation problem, we compute three different descriptors: a color histogram, an edge direction histogram and wavelet statistics. The features used have been selected for three basic properties: perceptual similarity (the feature distance between two images is large only if the images are not ‘‘similar’’), efficiency (the features can be rapidly computed) and economy (small dimensions that do not affect efficacy). The use of these assorted visual descriptors provides a

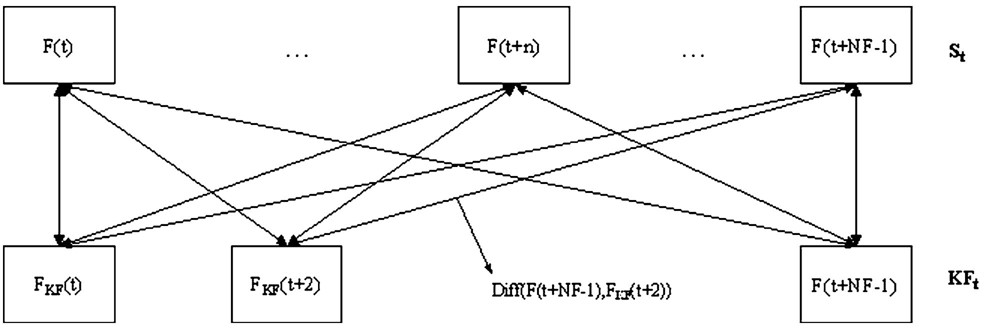


Fig. 2 Computation of the Fidelity measure. Each frame in the video sequence is compared with each key frame in the summary, and the minimal distance of each is retained. The greatest of these distances provides the Fidelity measure

more precise representation of the frame and captures slight variations between the frames in a shot.

Color histograms are frequently used to compare images because they are simple to compute and tend to be robust regarding small changes in camera viewpoint. Retrieval using color histograms to identify objects in image databases has been investigated in [29, 30]. An image histogram *h*( ) refers to the probability mass function of image intensities. Computationally, the color histogram is formed by counting the number of pixels belonging to each color. Usually a color quantization phase is performed on the original image in order to reduce the number of colors to consider in computing the histogram and thus the size of the histogram itself. The color histogram we use is composed of 64 bins determined by sampling groups of meaningful colors in the HSV color space [31]. The use of the HSV color space allows us to carefully define groups of colors in terms of Hue, Saturation and Lightness.

The edge direction histogram is composed of 72 bins corresponding to intervals of 2.5°. Two Sobel filters are applied to obtain the gradient of the horizontal and vertical edges of the luminance frame image [32]. These values are used to compute the gradient of each pixel; those pixels that exhibit a gradient above a predefined threshold are taken to compute the gradient angle and then the histogram. The threshold has been heuristically set at 4% of the gradient maximum value in order to remove from the histogram computation edges derived from background noise.

Multiresolution wavelet analysis provides representations of image data in which both spatial and frequency information are present [33]. In multiresolution wavelet analysis we have four bands for each level of resolution resulting from the application of two filters, a low-pass filter (L) and an high-pass filter (H). The filters are applied in pairs in the four combinations, LL, LH, HL and HH, and followed by a decimation phase that halves the resulting image size. The final image, of the same size as the original, contains a smoothed version of the original image (LL band) and three bands of details (see Fig. 5a).

Each band corresponds to a coefficient matrix that can be used to reconstruct the original image. These bands contain information about the content of the image in terms of general image layout (the LL band) and details (edges, textures, etc.). In our procedure the features are extracted from the luminance image using a three-step Daubechies multiresolution wavelet

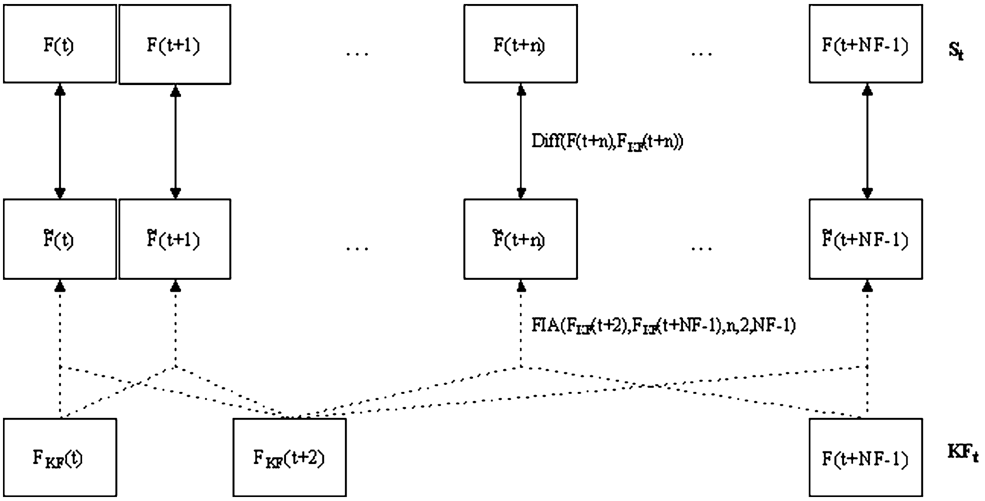


Fig. 3 The computation of the SRD measure. The video sequence is reconstructed using the key frames extracted and a frame interpolation algorithm (FIA). The interpolated frames are compared with the corresponding frames in the original sequence, and the computed frame differences are used to compute the final SRD measure

decomposition that uses 16 coefficients and producing 10 sub-bands [34] (Fig. 5b). Two energy features, the mean and standard deviation of the coefficients, are then computed for each of the 10 sub-bands obtained, resulting in a 20-valued descriptor.

To compare two frame descriptors, a difference measure is used to evaluate the color histograms, wavelet statistics and edge histograms. There are several distance formulas for measuring the similarity of color histograms. Techniques for comparing probability distributions are not appropriate because it is visual perception that determines the similarity, rather than the closeness of the probability distributions. One of the most commonly used measures is the histogram intersection [29]. The distance between two color histograms (d_H) using the intersection measure is given by

$$d_H(H_t, H_{t+1}) = 1 - \sum_{j=0}^{63} \min\left(H_t(j), H_{t+1}(j)\right), \tag{10}$$

where H_t and H_{t+1} are the color histograms for frame $F(t)$ and frame $F(t+1)$, respectively.

The difference between two edge direction histograms (d_D) is computed using the Euclidean distance as such in the case of two wavelet statistics (d_W):

$$\begin{aligned} d_D(D_t, D_{t+1}) &= \sqrt{\sum_{j=0}^{71} (D_t(j) - D_{t+1}(j))^2}, \\ d_W(W_t, W_{t+1}) &= \sqrt{\sum_{j=0}^{19} (W_t(j) - W_{t+1}(j))^2}, \end{aligned} \tag{11}$$

where D_t and D_{t+1} are the edge direction histograms and W_t and W_{t+1} are the wavelets statistics for frame $F(t)$ and frame $F(t+1)$.

The three resulting values (to simplify the notation we have indicated them as d_H, d_W and d_D only) are mapped into the range [0, 1] and then combined to form the final frame difference measure (d_{HWD}) as follows:

$$d_{\mathrm{HWD}} = (d_H d_W) + (d_W d_D) + (d_D d_H). \tag{12}$$

The aim of the frame difference measure is to accentuate dissimilarities in order to detect changes within the frame sequence. At the same time it is important that only when the frames are very different, the measure should report high difference values. As told before, the majority of the key frame selection methods exploit just one visual feature which is not sufficient to effectively describe an image content. If we were to use, for example, only the color histogram, a highly dynamic sequence (e.g., one containing fast moving or panning effects) with frames of the same color contents would result in a series of similar frame difference values and the motion effects would be lost. Similarly, frames with the same color content but different from the point of view of other visual attributes are considered similar. The uses of multiple feature can overcome these issues but pose the problem of their combination. In content-based retrieval systems, the features are combined by weighing them with suitable factors which are usually task-dependent [31]. We choose instead to use a different approach: the explicit selection of weight factors is removed by weighing each difference against the other. Moreover, this allows us to register significant differences in the d_{HWD} values only if at least two of the single differences exhibit high values (and thus two of the visual attributes emphasize the frame dissimilarity).

4.2 Key frame selection

The key frame selection algorithm that we propose dynamically selects the representative frames by analyzing the complexity of the events depicted in the shot in terms of pictorial changes. The frame difference values initially obtained are used to construct a curve of the cumulative frame differences which describes how the visual content of the frames changes over the entire shot, an indication of the shot's complexity:

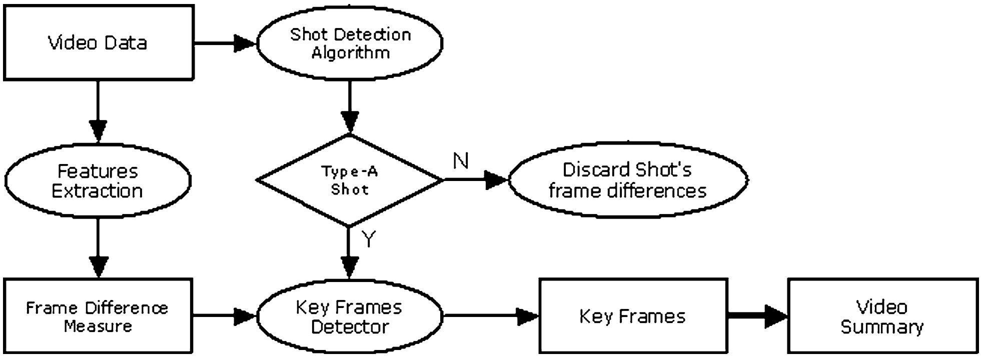


Fig. 4 The key frame extraction algorithm

sharp slopes indicate significant changes in the visual content due to a moving object, camera motion or the registration of a highly dynamic event. These cases must be taken into account in selecting the key frames to include in the shot summary. They are identified in the curve of the cumulative frame differences as those points at the sharpest angles of the curve (curvature or corner points). The key frames are those corresponding to the midpoints between each pair of consecutive curvature points. To detect the high curvature points we use the algorithm proposed by Chetverikov and Szabo [35]. The algorithm was originally developed for shape analysis in order to identify salient points in a 2D shape outline. The high curvature points are detected in a two-pass processing. In the first pass the algorithm detects candidate curvature points. The algorithm defines as a "corner" a location where a triangle of specified size and opening angle can be inscribed in a curve. Using each curve point P as a fixed vertex point, the algorithm tries to inscribe a triangle in the curve and then determines the opening angle $\alpha(P)$ in correspondence to P. Different triangles are considered using points that fall within a window of a given size w centered in P; the sharpest angle is retained as a possible high curvature point. This procedure is illustrated in Fig. 6. Defining the distance between points P and O as d_{PO}, the distance between points P and R as d_{PR} and the distance between points O and P as d_{OP}, the opening angle α corresponding to the triangle OPR is computed as

$$\alpha = \arccos \frac{d_{OP}^2 + d_{PR}^2 - d_{OR}^2}{2d_{OP}d_{PR}}. \tag{13}$$

A triangle satisfying the constraints on the distances between points (we consider only the x-coordinates):

$$\begin{aligned} d_{\min} &\leq |P_x - O_x| \leq d_{\max}, \\ d_{\min} &\leq |P_x - R_x| \leq d_{\max}, \end{aligned} \tag{14}$$

and the constraint on the angle values

$$\alpha \leq \alpha_{\max} \tag{15}$$

is called an admissible triangle. The first two constraints represent the operating window; the set of points contained in it are used to define the triangles. The third constraint is used to discard angles that are too flat. The sharpest opening angle of the admissible triangles is then assigned to P:

$$\alpha(P) = \min_{\alpha} \left\{ \alpha = O\hat{P}R \right\}. \tag{16}$$

If a point has no admissible triangles, the point is rejected assigning it an angle default value of π. In the second pass, those points in the set of the candidate high curvature points that are sharper than their neighbors (within a certain distance) are classified as high curvature points. A candidate point P is discarded if it has a sharper valid neighbor N, i.e., if

$$\alpha(P) \tag{17}$$

A point N is defined to be a neighbor of P if the following constraint is valid:

$$|P_x - N_x| \leq d_{\max}. \tag{18}$$

In our implementation we have defined the minimum points distance $d_{\min}$ as always equal to 1; consequently the only two parameters that influence the results of the algorithm are $d_{\max}$ and $\alpha_{\max}$. The most important parameter is $\alpha_{\max}$ which controls the set of admissible angles: a high value of $\alpha_{\max}$ will result in more points included in the set of candidate high curvature points, while a lower value indicates that only very sharp angles must be considered. This is the same as considering worthy of attention only slopes corresponding to sharp changes in the curve of the cumulative d_{HWD} frame differences.

Once the high curvature points have been determined, key frames can be extracted by taking the

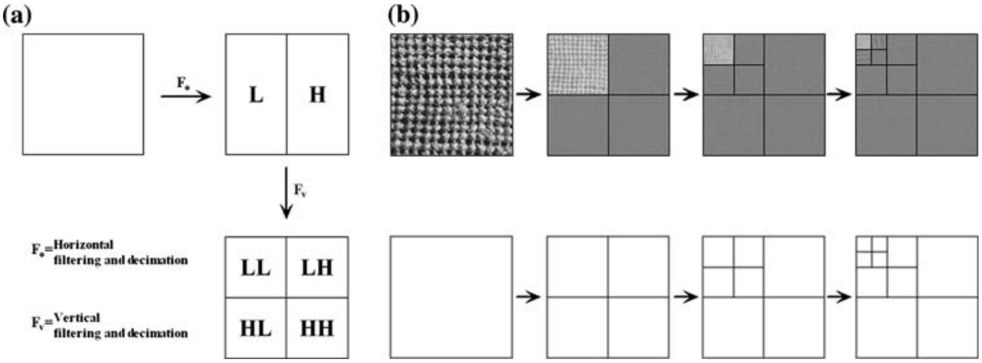

(a)
F_o
L
H
F_v
LL
LH
HL
HH
F_o=Horizontal filtering and decimation
F_v=Vertical filtering and decimation
(b)

Fig. 5 Multiresolution wavelet analysis. **a** The filtering and decimation of the image along the horizontal and vertical directions. Four bands are created each a quarter the size of the whole image. **b** The tree-step application of the multiresolution wavelet. The wavelet filters are applied to the top left band containing the resized image

midpoint between two consecutive high curvature points. Figure 7 is an example of how the algorithm works. The top image shows an example shot: the algorithm detects a high curvature point within the curve of cumulative frame differences. The first and last frames of the shot are implicitly assumed to correspond to high curvature points. The frames corresponding to the midpoints between each pair of consecutive high curvature points are selected as key frames (Fig. 7 center, triangles represent the high curvature points detected while the circles represent the key frames selected). If a shot does not present a dynamic behavior, i.e., the frames within the shot are highly correlated, the curve does not show evident curvature points, signifying that the shot can be summarized by a single representative frame. Figure 7 bottom shows the key frames extracted from the example shot. The summary contains the relevant elements of the frame sequence in terms of low-level features. Unlike some methods, such as those that extract key frames based on the length of the shots, our algorithm does not have to process the whole video. Another advantage is that it can extract the key frames on the fly: to detect a high curvature point we can limit our analysis to a fixed number of frame differences within a predefined window. Consequently the curvature points can be determined while computing the frame differences, and the key frames extracted as soon as a second high curvature point has been detected.

The high curvature points analysis for key frames extraction is similar to the approach proposed in [36] and also used in [27] where a polygonal curve representing the frames evolution is iteratively simplified by removing curve points. Unlike this approach, which requires that all the curve points should be globally analyzed at each step in order to select the candidate point to be removed, our detection of high curvature points can be made sequentially and locally.

As stated above, the corner point algorithm has two parameters: the size of the window within which the curvature angles are computed ($d_{\max}$) and the maximum value of the angle considered in determining the point's curvature angle ($\alpha_{\max}$). We have found experimentally that a size 3 window and angles of less than 140° provide a fair tradeoff between the number of computations required and the number of key frames extracted from each shot.

It is interesting to note that, in theory, the shot segmentation phase is not strictly required. Suppose that a segmentation algorithm is not available, and thus the video is a single sequence of frames. As cuts are abrupt changes in the visual content of the video sequence, our key frame selection algorithm can still detect them as corner points. The key frames extracted are the same as those extracted when the video is segmented. In case of fading or dissolving effects, the key frame extraction algorithm is not able to detect them since their visual evolution does not present sharp changes and they will not have corner point assigned. Thus, the selection of key frames within these gradual transition sequences cannot be entirely avoided. Take, for example, the case of the video that starts with a fade-in followed immediately by a cut: a key frame will be selected in the set of frames corresponding to the fade sequence. Consequently the segmentation algorithm serves to remove these kinds of shots, improving the summarization results by ensuring the selection of informative frames only.

5 Experimental setup

5.1 Algorithms tested

We have compared the results of our algorithm with those of five other key frame extraction algorithms. Together with our Curvature Points (CP) algorithm, we tested the Adaptive Temporal Sampling (ATS) algorithm of Hoon et al. [18], the ''Flexible Rectangles'' (FR) algorithm of Hanjalic et al. [17], the Shot Reconstruction Degree Interpolation (SRDI) algorithm of Liu Tieyan et al. [27], the Perceived Motion Energy (PME) algorithm of Liu Tianming et al. [21] and a simple Mid Point (MP) algorithm.

Both the ATS and FR algorithms select the key frames on the basis of the cumulative frame differences: they compute the color histogram differences on the RGB color space and plot them on a curve of cumulative differences. The key frames are selected by

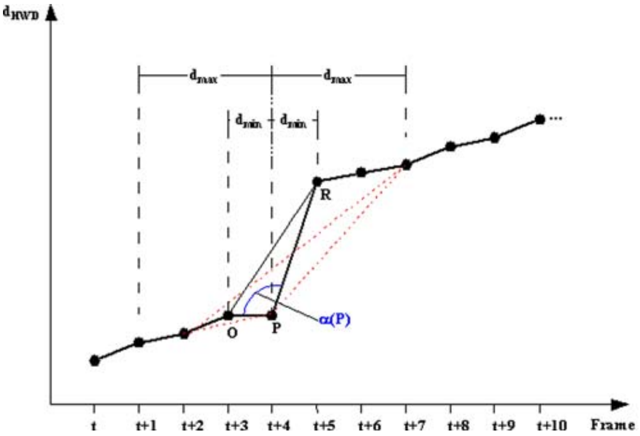


Fig. 6 Detection of the high curvature points algorithm

sampling the y-axis of the curve of the cumulative differences at constant intervals. The corresponding values in the x-axis represent the key frames. More key frames are likely to be found in intervals where the frame differences are pronounced than in intervals with lower values for frame differences. The FR algorithm also uses color histogram differences to build its curve, but the histograms are computed on the YUV color space. The curve of cumulative differences is approximated by a set of rectangles, each of which is used to select a key frame. As the widths of the rectangles are calculated to minimize the approximation error, an optimization algorithm is required (an iterative search algorithm is used). The input parameter of the algorithm is the number of key frames (and thus of the approximation rectangles) to be extracted. These

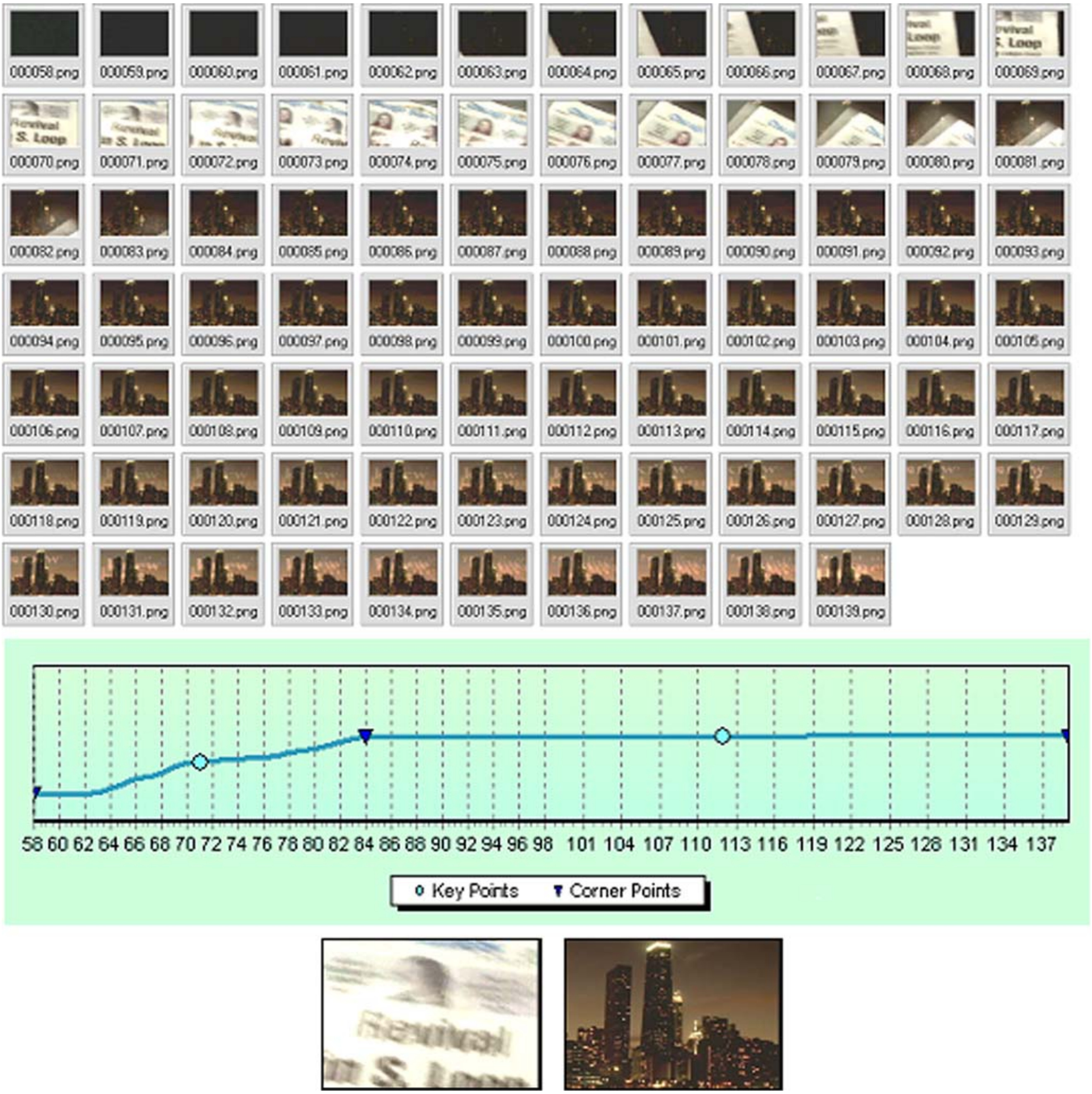


Fig. 7 An example of key frame selection with the proposed method. *Top* the shot to analyze. *Center* the corner points detected (*triangles*) and the key frames selected (*circles*). *Bottom* the two key frames, number 71 and 112, extracted from the example shot are shown

authors also propose a strategy for deciding how many key frames to select from the shots of the video sequence: they assign the number of key frames in proportion to the length of the shot.

Given a specified number of key frames to be extracted, the SRDI algorithm uses the motion vectors to compute the frame's motion energy. All the motion energy values are then used to build a motion curve that is passed to a polygon simplification algorithm. This algorithm retains only the most salient points that can approximate the whole curve. The frames corresponding to these points form the key frame set. If the number of frames in the final set differs from the number of key frames requested, the set is reduced or increased by interpolating frames according to the SRD criteria. When the number of frames is lower than the number desired, the shot is reconstructed by interpolating the frames in the frame set, and the interpolated frames that have largest reconstruction errors are retained up to the number of key frames needed. When the number of frames in the frame set is greater than the number of key frames needed, the frames in the frame set are interpolated, and those with the minimal reconstruction error are removed from the set. The result of the SRDI algorithm depends on both the polygon simplification algorithm and the SRD criteria. The only parameter that must be set for the ATS, FR and SRDI algorithms is the number of key frames to be extracted.

The PME algorithm works on compressed video using the motion vectors as indicators of the visual content complexity of a shot. A triangle model based on a PME feature is used to select key frames. With this model, a shot is segmented into sub-segments representing an acceleration and deceleration motion pattern (each modeled by a triangle). Key frames are then extracted from the shots by taking the vertices of the triangles. To compute the PME feature, the magnitudes of the motion vectors of the B frames are first filtered with two nonlinear filters. For each motion vector in the frame feature, a spatial filter is applied within a given spatial window, and a temporal filter is applied on values belonging to frames within a given temporal window. For each B frame, the PME is then computed on the magnitudes of the motion vectors and the dominant motion direction. This preprocessing requires the setting of several parameters.

A simple procedure then automatically computes the triangles on the PME values and the corresponding key frames. The algorithm requires the setting of two parameters, the most important of which is the minimum size of a triangle as it influences the length of the interval between two consecutive key frames. The MP algorithm was chosen because it can represent the extreme case of our algorithm, when no evident high curvature points can be found in a shot and the center frame of the sequence is chosen as the key frame instead.

Where available the parameters set for the algorithms were always those reported in the original papers. The ATS, FR and SRDI algorithms require the input parameter of the number of key frames that must be provided. Defining a general rule for setting this number is a crucial matter; the results may vary widely, depending on the rule selected. We have set the input parameter for these algorithms as the same number of key frames found by our algorithm. We can then compare the algorithms regardless of the number of key frames: any difference in results depends only on the selection strategy adopted. Because the PME algorithm, instead, extracts the key frames in a totally automatic way, as does our algorithm, the results depend on both the number of key frames extracted and the selection strategy applied.

5.2 Video data set

Six videos of various genre were used to test the performance of the key frame extraction algorithms. Table 1 summarizes the characteristics of the six video test sequences. The ‘‘eeopen'' video is a MPEG1 introduction sequence of a TV series with short shots and several transition effects. The ‘‘news'' and ‘‘nwanw1'' are two MPEG1 news sequences; the shots are moderately long, not too dynamic and mixed with commercial sequences of very fast-paced shots. The ‘‘nwanw1'' video is similar

Table 1 The six videos used to test the key frame extraction algorithms

Video name	Genre	Length (mm:ss)	Resolution ($W\times H$)	TNF	NS
eeopen	TV series intro	00:42	352×240	1,289	24
nwanw1	News with commercials	03:39	176×112	6,556	39
news	News with commercials	02:39	176×112	4,757	12
football	Sport	03:43	176×112	6,697	28
bugsbunny	Short cartoon	07:30	352×240	13,492	89
basketball2	Single shot sport sequence	00:30	320×220	893	1

TNS total number of frames, *NS* number of shots found. Both refer to type A and type B shots to the "news" video, but has longer shots. The "football" and "basketball2" videos are two MPEG1 sport sequences: "football" exhibits rather long shots, while "basketball2" is a single long shot, and both have panning and camera motion effects. Finally, "bugsbunny" is a MPEG1 short cartoon sequence with many shots and a number of transition effects.

The videos were chosen in order to evaluate the capability of the key frame extraction algorithms to cope with shots of different length and with various levels of dynamic events captured. In particular the single shot of the "basketball2" video was chosen to test the ability of the algorithms to effectively capture the content of the long sequence in the presence of considerable camera motion. Each video was processed by each of the six key frame extraction algorithms. The Fidelity and the SRD measure were computed on the resulting summaries. The parameters applied were not changed when passing from one video to the other, except for the number of key frames extracted for the ATS, FR and SRDI algorithms.

5.3 Measures setup

Since key frames are extracted from each shot, we must take this into account when computing the quality measures defined in Sect. 3. For each shot of type A the Fidelity and the SRD measure are computed; if an algorithm does not extract key frames from the shot, the corresponding quality measures are set at equal to the worst case value (zero for both Fidelity and SRD). The final quality measure for the entire video is then computed as the average of all Fidelity and SRD measures for the shots:

$$\begin{aligned} \text{Fidelity} &= {}^{1}\!/_{\gamma_{\text{NS}}} \sum_{t=1}^{\gamma_{\text{NS}}} \text{Fidelity}(\mathbf{S}_t, \mathbf{KF}_t), \\ \text{SRD} &= {}^{1}\!/_{\gamma_{\text{NS}}} \sum_{t=1}^{\gamma_{\text{NS}}} \text{SRD}(\mathbf{S}_t, \mathbf{KF}_t), \end{aligned} \tag{19}$$

where γ_{NS} is the number of shots. For both measures the higher the value the better the summary represents the video content. The compression ratio measure was computed considering the whole video as follows:

$$\text{CRatio} = 1 - {}^{\gamma_{\text{TNKF}}}\!/_{\gamma_{\text{TNF}}}, \tag{20}$$

where γ_{TNKF} is the total number of key frames extracted and γ_{TNF} the total number of frames in shots of type A.

Both the Fidelity and the SRD measure require that a frame difference be defined. In order to evaluate the influence of frame difference on the quality measures, the Fidelity and the SRD were computed using two different formulations, the first is based on the previously defined color histogram differences computed with the histogram intersection formula, the second on the HWD measure instead.

Since the number of key frames extracted from a shot is usually small (even one only), to compute the SRD we opted for a simple linear interpolation algorithm on the frame's descriptive features. One reason for this choice is that a more complex interpolation algorithm cannot be used effectively when the interpolation points at disposal are very few. Another reason is that pixel values depend on the objects moving in the scene, and interpolating each pixel of the frame using only the color information may result in a very noisy image that does not reflect the real content of the original frames. Interpolating global features instead can capture the overall pictorial content of the frames without having to take motion into account. To allow frame interpolation when only one key frame was available we also selected the first and last frames of the shot as interpolation points.

6 Experimental results

Table 2 summarizes some of the characteristics of the algorithms tested. The top row in Table 2 regards the most important property of a key frame selection algorithm: it must extract key frames in a totally automatic way without requiring that the user specify the number of key frames to be extracted as a parameter. Otherwise the user should know the video content in advance and adjust the parameter accordingly. The next row indicates that the algorithm should be flexible enough to extract a variable number of key frames according to some criteria (e.g., shot complexity, shot dynamics or shot length). By "on-the-fly processing" we mean the ability of the given algorithm to output key frames without having to process all the frames in the shot or, worse, all the frames in the video sequence. By "real-time processing" we mean mainly that the time the algorithm takes to compute the key frames (with all the eventual pre-processing phases included) must be less than the time required to view the whole video sequence. Some algorithms require that motion vectors video (usually from MPEG-compressed video) be provided. These algorithms do not decode the frames but their applicability is limited to a specific compression standard algorithm: videos that are not compressed would have to be compressed before being processed.

Table 2 Comparison of the six key frame extraction algorithms used in our experiment

	ATS	FR	SRDI	MP	PME	CP
Automatic key frames selection	N	N	N	Y	Y	Y
Variable number of key frames	Y	Y	Y	N	Y	Y
On-the-fly processing	N	N	N	Y	N	Y
Real-time processing	Y	Y	N	Y	N	Y
Requires motion vectors	N	N	Y	N	Y	N
Uses an optimization algorithm	N	Y	N	N	N	N
Shot length sensitive	?	Y	N	N	Y	N
Reference	[18]	[17]	[26]	–	[21]	

For each algorithm some important characteristics are reported

On the contrary, algorithms that work on uncompressed data, when presented with a compressed video, need to decode the frames. Due to the asymmetric nature of the compression algorithms, the coding time is usually much higher than the decoding time. Algorithms that implement an optimization algorithm usually offer better results, but present the drawback of being time consuming and needing efficient implementation in order to be practical [24]. "Shot length sensitive" refers to the fact that the key frames extracted depend in some way on the length of a shot. This is, for example, the case of the FR algorithm, where the number of key frames is proportional to the length of a given shot or that of the PME, where the algorithm's parameter defines the minimum gap between two consecutive key frames. In case of the ATS algorithm, no evaluation can be made since the original publication gives no indication of how the key frames are distributed among the shots.

The motion vectors for the SRDI and PME algorithms have been directly extracted from the compressed video. The ATS, FR, CP and the interpolation of the SRDI algorithm have been performed on sub-sampled frames. By experiments we have observed that we can sub-sample the frames down to 48 pixels (with respect to the larger dimension and maintaining the aspect ratio) and still obtaining similar results as the application of the algorithms to the original frames. To limit the risk to lose information, especially for the edge histogram and the wavelet statistics, we have chosen to sub-sample the frames to a dimension of 64 pixels.

6.1 Theoretical complexity

Table 3 shows the theoretical complexity of the key frame extraction algorithms. We have used the same approach and notation in Lefevre et al. [37]. The complexity was computed considering mathematical, logical and comparison operations (all supposed having cost one). We have not taken into account memory usage, allocation, de-allocation or the cost required to decode a frame. In computing the complexity we have considered the cost required to compute the features needed by the algorithm (such as color histograms, statistics, etc.), the cost required to compute the values to be analyzed (such as the cumulative frame differences or the PME values) and the cost required by the actual key frame detection algorithm. All the costs were computed considering the algorithms' parameters. Following the authors notation,N is the number of bins in a histogram (we have used the same symbol to indicate the size of a feature vector),P the number of pixels in a frame, andB the number of blocks of pixels in a frame (e.g., the macro-blocks in a compressed video). Finally, we indicate withK the number of key frames. The complexity is relative to the operations per frame required to process a shot.

Although the key frame selection step of the PME algorithm is simple, the pre-processing phase performed with the two nonlinear filters penalizes the algorithm: more data must be analyzed, and each step of the filtering process also requires data reordering before the final result for that step can be obtained. More than half of the operations required for the CP algorithm are due to the wavelet computation (we have used the standard wavelet decomposition). Using the lifting scheme will reduce the complexity by approximately half [38]. The # annotation for the SRDI algorithm indicates that the one-time operations performed on the key frames (divided for the number of frames) for the background classification have not been added. The + annotation for the SRDI and PME algorithms indicates instead that the complexity reported does not include the motion vectors computation. The SRDI and PME use motion vectors and thus they can take advantage of compressed video using them. If the motion vectors are not available an additional cost should be added for each frame requiring forward or backward motion vectors com-

Table 3 Complexities of the key frame algorithms tested

Algorithm	Complexity
ATS	$O(3P + 9N)$
FR	$O(18P + 9N + 2K)$
CP	$O(72P + 13N)$
SRDI	$O(53P)^{\#}_{+}$
PME	$O(958B)_{+}$
MP	$O(2)$

putation. For example, using the fast 2D log search algorithm, which is sub-optimal, requires an additional $O(75\mathbb{P})$ operations [37].

6.2 Computational time

Table 4 shows the computation time (in minutes, seconds and thousands of seconds) of the algorithms tested including the frame decoding time for the algorithms that required it. We have implemented the key frames extraction algorithms in C++ under the Borland C++ Builder 5.0 development environment with the default optimization (for faster execution) turned on. The computer used for the comparison was an AMD Athlon 1 GHz with 512 MB of RAM and running a Windows 2000 Professional operating system. For image processing algorithms, separable filters were used whenever it was possible. Results that can be used in subsequent operations were computed only once. A test session was performed by processing all the six video sequences with a key frame extraction algorithm. Before the next test session the system was restarted. For each video the computational time reported refers to the average time of three test sessions. The MP algorithm is not included since its computation is virtually instantaneous. The algorithm's computational time is reported in relative form with respect to the ATS computational time.

For the apparent discrepancies in the computational time of the ATS and FR algorithms with respect to the CP algorithm, it should be noted that the decoding time take up a significant part of the processing time of the two algorithms. For the "eeopen", "bugsbunny" and "basketball" videos, the decoding time represents about 90% (ATS) and 82% (FR) of the total computational time. For the remaining three videos, the decoding time represents 70 and 50% of the ATS and FR total time, respectively. The high computational time of the SRDI and PME algorithms is mainly due to the heavy preprocessing required before the key frame can be selected. The SRDI algorithm requires pixel interpolation based on motion vectors and pixel background classification. Although the key frame selection step of the PME algorithm is simple, the pre-processing phase performed with the two nonlinear filters penalizes the algorithm: more data must be analyzed, and each step of the filtering process also requires data reordering before the final result for that step can be obtained.

Table 4 Computational time of the five algorithms on the six videos

Video	ATS	FR	CP	SRDI	PME
eeopen	00:10:596	1.01	2.86	3.97	9.03
nwanw1	00:14:152	1.35	5.75	9.16	7.22
news	00:10:425	1.34	5.74	9.09	6.66
football	00:14:531	1.30	5.81	8.78	7.68
bugsbunny	01:38:542	1.13	2.98	4.65	9.41
basketball	00:06:570	1.08	3.08	4.55	6.97

The times reported for the ATS algorithm are in minutes, seconds and thousands of seconds. For the other algorithms, the times are in relative form with respect to the ATS

6.3 Results

In our preliminary experiments, the Fidelity and SRD measures for the FR algorithm were more than 20% lower than those for the other algorithms. As stated before, in its original formulation, the FR algorithm assigns to each shot a number of key frames proportional to the length of the shot. When the total number of key frames to be extracted is comparable to the number of shots, there is a high probability that no key frames will be assigned to some of the shots. Thus for a fair comparison, in subsequent experiments we have modified the assignment criteria for the FR algorithm. In this modified formulation, the FR algorithm extracts from each shot the same number of key frames computed by the CP algorithm. The same criterion was applied to the ATS algorithm.

Tables 5, 6 and 7 show the results of the different algorithms on the six video sequences. Table 5 shows the compression ratio (CRatio) values and some statistics. KFPerShot indicates the average number of key frames for each shot: nUsedShots and FrPerShots indicate the number of type A shots and the average number of frames per shots, respectively. From Table 5 we can see that for the "eeopen", "nwanw1" and "bugsbunny" videos some type B shots have been removed by our shot detection algorithm. Our key frame selection algorithm is able to extract more than one frame per type A shot as is the PME algorithm (that the number of key frames of the FR, ATS and SRDI algorithms depends on the CP algorithm). Videos with high dynamics, in fact, are assigned more key frames per shot than those with little motion. This can be seen in the case of "basketball2", "football" and "news" videos. For the "bugsbunny" video, although very few shots exhibit high dynamics, the PME algorithm extracts nearly double the number of key frames than the CP algorithm.

Table 6 gives the Fidelity measure results with the minimum and standard deviation of the measure

Table 5 Compression ratios (CRatio) and key frames statistic results of the algorithms on the six video sequences

	CRatio	Key frames	KFPerShot	nTotShots	nUsedShots	FrPerShot
eeopen						
CP	0.980	25	1.136	24	22	53.500
FR	0.980	25	1.136	24	22	53.500
ATS	0.980	25	1.136	24	22	53.500
SRDI	0.980	25	1.136	24	22	53.500
PME	0.971	36	1.636	24	22	53.500
MP	0.982	22	1.000	24	22	53.500
nwanw1						
CP	0.991	61	1.743	39	35	168.256
FR	0.991	61	1.743	39	35	168.256
ATS	0.991	61	1.743	39	35	168.256
SRDI	0.991	61	1.743	39	35	168.256
PME	0.990	66	1.886	39	35	168.256
MP	0.995	35	1.000	39	35	168.256
news						
CP	0.993	35	2.917	12	12	397.083
FR	0.993	35	2.917	12	12	397.083
ATS	0.993	35	2.917	12	12	397.083
SRDI	0.993	35	2.917	12	12	397.083
PME	0.994	28	2.333	12	12	397.083
MP	0.997	12	1.000	12	12	397.083
football						
CP	0.988	82	2.929	28	28	239.286
FR	0.988	82	2.929	28	28	239.286
ATS	0.988	82	2.929	28	28	239.286
SRDI	0.988	82	2.929	28	28	239.286
PME	0.992	55	1.964	28	28	239.286
MP	0.996	28	1.000	28	28	239.286
bugsbunny						
CP	0.993	95	1.145	89	83	151.551
FR	0.993	95	1.145	89	83	151.551
ATS	0.993	95	1.145	89	83	151.551
SRDI	0.993	95	1.145	89	83	151.551
PME	0.986	182	2.193	89	83	151.551
MP	0.994	83	1.000	89	83	151.551
basketball2						
CP	0.983	15	15.000	1	1	894.000
FR	0.983	15	15.000	1	1	894.000
ATS	0.983	15	15.000	1	1	894.000
SRDI	0.983	15	15.000	1	1	894.000
PME	0.993	6	6.000	1	1	894.000
MP	0.999	1	1.000	1	1	894.000

KFPerShot the average number of key frames for each shot, *nUsedShots* the number of type A shots, *FrPerShots* the average number of frames per shot

computed on both the histogram and the HWD descriptors. Table 7 presents the results of the SRD measure. They show that the average Fidelity and SRD measures computed on the histogram (HST) are less variable than those computed on the HWD. This seems to indicate that frame differences tend to be less distinguishable when using the color histogram alone. It is also interesting to note that the summaries of the “basketball2” video are clearly unacceptable if we evaluate them with the Fidelity measure computed on the histogram, while they appear fairly acceptable if evaluated with the HWD.

To judge the performance of our algorithm with respect to the other five algorithms, it is useful to express the results as a measure of relative improvement (ΔQ) using the following formula:

$$\Delta Q(X)=\frac{(\mathrm{Measure\_Alg(CP)}-\mathrm{Measure\_Alg}(X))}{\mathrm{Measure\_Alg}(X)}, \tag{21}$$

where Measure_Alg corresponds to the Fidelity and the SRD measure, and we substitute X with FR, ATS, SRDI, PME and MP in turn.

Table 8 details for each video the relative edge of the performance of our algorithm over that of other five algorithms. It can be seen that when the number of key frames per shot is small as in the case of the “eeopen” and “bugsbunny” videos (1,136 and 1,145 key

Table 6 Fidelity measure results computed using the histogram (HST) and the HWD descriptors

	Fid. HST	Fid. HST (mi)	Fid. HST (SD)	Fid. HWD	Fid. HWD (mi)	Fid. HWD (SD)
eeopen						
CP	0.762	0.069	0.228	0.907	0.217	0.185
FR	0.760	0.072	0.232	0.906	0.184	0.191
ATS	0.756	0.016	0.236	0.885	0.123	0.235
SRDI	0.761	0.077	0.236	0.899	0.298	0.184
PME	0.763	0.072	0.228	0.892	0.133	0.213
MP	0.770	0.069	0.207	0.901	0.184	0.203
nwanw1						
CP	0.642	0.013	0.325	0.778	0.192	0.310
FR	0.631	0.014	0.339	0.758	0.110	0.345
ATS	0.645	0.014	0.323	0.783	0.113	0.308
SRDI	0.619	0.046	0.343	0.762	0.035	0.313
PME	0.620	0.017	0.334	0.745	0.199	0.352
MP	0.613	0.007	0.354	0.733	0.015	0.371
news						
CP	0.607	0.166	0.258	0.860	0.662	0.125
FR	0.614	0.167	0.250	0.870	0.656	0.117
ATS	0.612	0.194	0.248	0.872	0.671	0.115
SRDI	0.578	0.142	0.278	0.851	0.656	0.129
PME	0.600	0.139	0.275	0.851	0.648	0.137
MP	0.584	0.074	0.283	0.836	0.588	0.158
football						
CP	0.627	0.293	0.208	0.858	0.512	0.133
FR	0.627	0.354	0.202	0.842	0.377	0.154
ATS	0.642	0.331	0.196	0.858	0.537	0.133
SRDI	0.611	0.268	0.214	0.840	0.447	0.147
PME	0.612	0.254	0.213	0.825	0.431	0.167
MP	0.601	0.243	0.233	0.834	0.183	0.185
bugsbunny						
CP	0.656	0.052	0.239	0.907	0.118	0.148
FR	0.656	0.052	0.239	0.904	0.118	0.150
ATS	0.665	0.065	0.226	0.906	0.141	0.141
SRDI	0.647	0.024	0.245	0.904	0.283	0.134
PME	0.678	0.004	0.232	0.910	0.351	0.126
MP	0.650	0.052	0.247	0.901	0.118	0.152
basketball2						
CP	0.093	0.093	0.000	0.556	0.556	0.000
FR	0.094	0.094	0.000	0.528	0.528	0.000
ATS	0.064	0.064	0.000	0.528	0.528	0.000
SRDI	0.081	0.081	0.000	0.552	0.552	0.000
PME	0.075	0.075	0.000	0.533	0.533	0.000
MP	0.031	0.031	0.000	0.407	0.407	0.000

The minimum (mi) and standard deviation (SD) of the Fidelity measures computed on each shot are also reported

frames per shot, respectively) the differences between the CP, FR, ATS and MP algorithms are slight. Exceptions are the SRDI algorithm, about 5% worse than the CP algorithm, and the PME algorithm in the ‘‘bugsbunny’’ video, about 4% better than the CP algorithm (note that the PME algorithm extracts 182 key frames and the CP algorithm 95). For the other videos, the PME algorithm shows instead worst results (excluding the MP algorithm). As the number of key frames per shot increases, the gap is more marked. With the exception of the ‘‘basketball2’’ video, the performances of the FR algorithm and the CP algorithm do not exhibit large differences; only in the ‘‘news’’ video does the FR algorithm outperform (slightly) the CP algorithm. The most interesting experiment is the one regarding the ‘‘basketball2’’ video. The high dynamics and the length of the shot cause greatly diverging results. The performance of the SRDI algorithm exhibits a variable behavior depending on the measure employed, while the MP algorithm shows the worse results.

Table 9 lists the ΔQ measure of relative improvement computed as the percentage average for all five videos. Overall our algorithm outperforms the other five algorithms. Its advantages over the FR algorithm are negligible, but we must remember that the FR

Table 7 Shot Reconstruction Degree (SRD) measure results computed using the histogram (HST) and the HWD descriptors

	SRD HST	SRD HST (mi)	SRD HST (SD)	SRD HWD	SRD HWD (mi)	SRD HWD (SD)
eeopen						
CP	4.882	2.214	1.211	7.591	3.814	1.700
FR	4.868	2.077	1.239	7.577	3.741	1.723
ATS	4.660	2.072	1.246	7.259	3.729	1.757
SRDI	4.693	1.984	1.189	7.314	3.416	1.684
PME	4.949	1.752	1.501	7.550	3.308	2.065
MP	4.846	1.842	1.283	7.563	3.682	1.749
nwanw1						
CP	3.887	1.474	1.162	6.648	2.928	1.634
FR	3.887	1.369	1.212	6.632	2.696	1.690
ATS	3.849	1.527	1.199	6.588	2.984	1.664
SRDI	3.587	0.636	1.295	6.104	1.402	1.959
PME	3.581	1.280	1.255	6.067	2.668	1.839
MP	3.642	0.897	1.306	6.311	1.858	1.919
news						
CP	3.520	2.167	1.372	6.186	4.355	2.064
FR	3.534	2.122	1.374	6.238	4.296	2.048
ATS	3.515	2.006	1.350	6.188	4.317	2.033
SRDI	3.209	1.415	1.504	5.663	2.974	2.273
PME	3.084	1.714	1.189	5.527	3.279	1.993
MP	3.147	1.513	1.537	5.651	3.218	2.357
football						
CP	3.335	2.294	0.766	5.856	4.270	1.076
FR	3.361	2.294	0.727	5.902	4.270	1.008
ATS	3.276	1.847	0.844	5.684	3.522	1.196
SRDI	3.067	1.817	0.834	5.327	3.319	1.230
PME	2.846	1.634	0.816	5.049	3.026	1.248
MP	2.814	1.467	0.802	5.121	2.809	1.334
bugsbunny						
CP	3.442	1.176	1.091	6.476	2.898	1.479
FR	3.444	1.176	1.095	6.483	2.898	1.488
ATS	3.339	1.201	1.070	6.275	2.769	1.464
SRDI	3.268	0.865	1.137	6.134	1.243	1.656
PME	3.597	1.170	1.151	6.521	2.324	1.586
MP	3.388	1.176	1.128	6.410	2.898	1.533
basketball2						
CP	2.269	2.269	0.000	4.137	4.137	0.000
FR	2.079	2.079	0.000	4.044	4.044	0.000
ATS	1.980	1.980	0.000	3.786	3.786	0.000
SRDI	2.356	2.356	0.000	4.120	4.120	0.000
PME	1.658	1.658	0.000	3.336	3.336	0.000
MP	0.693	0.693	0.000	2.027	2.027	0.000

The minimum (mi) and standard deviation (SD) of the SRD measures computed on each shot are also reported

algorithm uses an optimization strategy to allocate the key frames within a shot and requires that the number of key frames must be given a priori. For the ATS and SRDI algorithms, the gap is more noticeable and, as the number of key frames extracted is the same, the results depend only on the selection strategies adopted by the algorithms. The poor performance of the MP algorithm is, of course, to be expected since it extracts only one key frame from each shot.

More interesting is the performance of the PME algorithm, which extracts key frames in a totally automatic way based on motion vectors. The results of the SRD measure show that the CP algorithm outperforms the PME algorithm by about 10%. The key frames selected only by motion do not seem to be successful in visually representing the content of the video.

7 An application for key frames

Our key frame extraction algorithm is currently being applied in the ‘‘Archivio di Etnografia e Storia Sociale—AESS’’ (Archive of Social History and Ethnography) [39]. The AESS archive has as its purpose the conservation, consultation and employment of documents and images regarding the life and social trans-

Table 8 The relative improvement (ΔQ) measured on the Fidelity and SRD results of the proposed algorithm with respect to each of the other algorithms tested

	Fidelity HST	Fidelity HWD	SRD HST	SRD HWD
eeopen				
CP	0.0	0.0	0.0	0.0
FR	0.3	0.1	0.3	0.2
ATS	0.8	2.5	4.8	4.6
SRDI	0.1	0.9	4.0	3.8
PME	–0.1	1.7	–1.4	0.5
MP	–1.0	0.7	0.7	0.4
nwanw1				
CP	0.0	0.0	0.0	0.0
FR	1.7	2.6	0.0	0.2
ATS	–0.5	–0.6	1.0	0.9
SRDI	3.7	2.1	8.4	8.9
PME	3.5	4.4	8.5	9.6
MP	4.7	6.1	6.7	5.3
news				
CP	0.0	0.0	0.0	0.0
FR	–1.1	–1.1	–0.4	–0.8
ATS	–0.8	–1.4	0.1	0.0
SRDI	5.0	1.1	9.7	9.2
PME	1.2	1.1	14.1	11.9
MP	3.9	2.9	11.9	9.5
football				
CP	0.0	0.0	0.0	0.0
FR	0.0	1.9	-0.8	–0.8
ATS	–2.3	0.0	1.8	3.0
SRDI	2.6	2.1	8.7	9.9
PME	2.5	4.0	17.2	16.0
MP	4.3	2.9	18.5	14.4
bugsbunny				
CP	0.0	0.0	0.0	0.0
FR	0.0	0.3	–0.1	–0.1
ATS	–1.4	0.1	3.1	3.2
SRDI	1.4	0.3	5.3	5.6
PME	–3.2	–0.3	–4.3	–0.7
MP	0.9	0.7	1.6	1.0
basketball2				
CP	0.0	0.0	0.0	0.0
FR	–1.1	5.3	9.1	2.3
ATS	45.3	5.3	14.6	9.3
SRDI	14.8	0.7	–3.7	0.4
PME	24.0	4.3	36.9	24.0
MP	200.0	36.6	227.4	104.1

The results are in percentages

Table 9 The average of the relative improvement (ΔQ) measured on the Fidelity and SRD results of the proposed algorithm with respect to each of the other algorithms tested

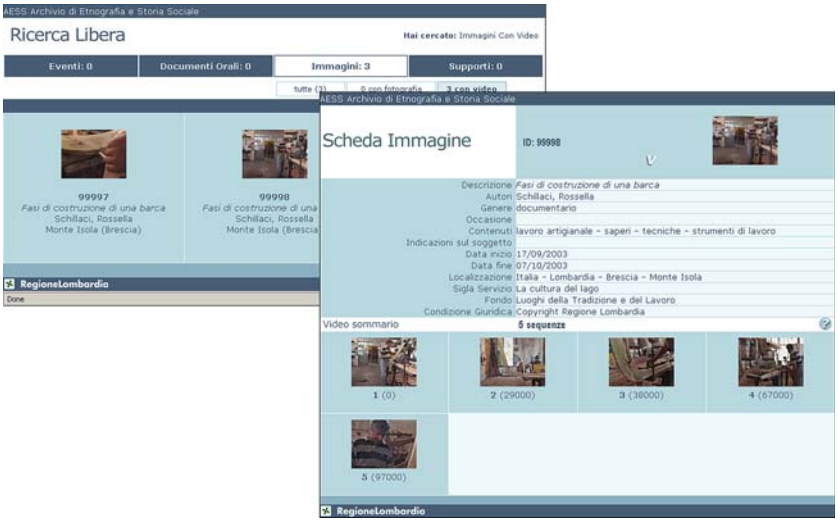
Overall	Fid. HST	Fid. HWD	SRD HST	SRD HWD
CP	0.0	0.0	0.0	0.0
FR	0.0	1.5	1.4	0.2
ATS	6.9	1.0	4.2	3.5
SRDI	4.6	1.2	5.4	6.3
PME	4.6	2.5	11.8	10.2
MP	35.5	8.3	44.5	22.4

The results are in percentages

formation, literature, oral history and the cultural artifacts of the Lombard region. The AESS Web site has been designed and implemented at the CNR–ITC Unit of Milan for the project promoted by the Lombard Region (the Direzione generale Culture, Identità e Autonomie della Lombardia) and Interreg II Internum project of the European Union. The archive of the AESS Web site stores information concerning the oral history of the Lombard region: it is mainly composed of popular songs and other audio and video records describing the popular traditions handed down generation by generation, such as traditional fairs, and customs. The images and videos represent occasions on which the audio has been performed as songs, recorded as interviews, etc. Linked with the audio and image data are books, journals, discs, DVD, etc., on which this information is stored (printed, recorded, etc.). Video information is handled with the collaboration of the DISCo department of the University of Milano-Bicocca. Figure 8 shows a page of the AESS Web site where the videos are stored and an example of a visual summary associated with a chosen video. Within the AESS Web site, the video key frames are used both as visual content abstracts for the users and as entry points to video sub-sequences.

8 Conclusion

In this paper, we have presented an innovative algorithm for key frame extraction. By analyzing the differences between pairs of consecutive frames of a video sequence, the algorithm determines the complexity of the sequence in terms of changes in visual content as expressed by different frame descriptors. Similarity measures are computed for each descriptor and combined to form a frame difference measure. The algorithm, which does not exhibit the complexity of existing methods based, for example, on clustering or optimization strategies, can dynamically and rapidly select a variable number of key frames within each sequence. Another advantage is that it can extract the key frames on the fly: key frames can be determined while computing the frame differences as soon as two high curvature points have been detected. The performance of this algorithm has been compared with that of other key frame extraction algorithms based on different approaches. The summaries have been evaluated objectively with three quality measures: the Fidelity measure, the Shot Reconstruction Degree measure and the compression ratio measure. Experimental results show that the proposed algorithm out-



AESS Archivio di Etnografia e Storia Sociale

Ricerca Libera

Hai cercato: Immagini Con Video

Eventi: 0	Documenti Orali: 0	Immagini: 3	Supporti: 0

tutte (3) 0 con fotografia 3 con video

99997
Fasi di costruzione di una barca
Schillaci, Rossella
Monte Isola (Brescia)

99998
Fasi di costruzione di una
Schillaci, Rossella
Monte Isola (Brescia

RegioneLombardia

Done

AESS Archivio di Etnografia e Storia Sociale

Scheda Immagine

ID: 99998

Descrizione	*Fasi di costruzione di una barca*
Autori	Schillaci, Rossella
Genere	documentario
Occasione	
Contenuti	lavoro artigianale - saperi - tecniche - strumenti di lavoro
Indicazioni sul soggetto	
Data inizio	17/09/2003
Data fine	07/10/2003
Localizzazione	Italia - Lombardia - Brescia - Monte Isola
Sigla Servizio	La cultura del lago
Fondo	Luoghi della Tradizione e del Lavoro
Condizione Giuridica	Copyright Regione Lombardia

Video sommario **5 sequenze**

1 (0)
2 (29000)
3 (38000)
4 (67000)
5 (97000)

RegioneLombardia

Fig. 8 The Web site of the AESS archive where the CP key frame algorithm is currently employed

performs the other key frame extraction algorithms investigated.

Acknowledgements The video indexing and analysis presented here was supported by the Italian MURST FIRB Project MAIS (Multi-channel Adaptive Information Systems) [40] and by the Regione Lombardia (Italy) within the INTERNUM project aimed at facilitating access to the cultural video documentaries of the AESS (Archivio di Etnografia e Storia Sociale).

References

1. Agraim, P., Zhang, H., Petkovic, D.: Content-based representation and retrieval of visual media: a state of the art review. Multimed. Tools Appl. **3**, 179–202 (1996)
2. Dimitrova, N., Zhang, H., Shahraray, B., Sezan, M., Huang, T., Zakhor, A.: Applications of video-content analysis and retrieval. IEEE MultiMed. **9**(3), 44–55 (2002)
3. Antani, S., Kasturi, R., Jain, R.: A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. Pattern Recognit. **35**, 945–965 (2002)
4. Schettini, R., Brambilla, C., Cusano, C., Ciocca, G.: Automatic classification of digital photographs based on decision forests. Int. J. Pattern Recognit. Artif. Intell. **18**(5), 819–846 (2004)
5. Fredembach, C., Schröder, M., Süsstrunk, S.: Eigenregions for image classification. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **26**(12), 1645–1649 (2004)
6. Hauptmann, A.G., Jin, R., Tobun, D.N.: Video retrieval using speech and image information. In: Proceedings of Electronic Imaging Conference (EI'03), Storage Retrieval for Multimedia Databases, Santa Clara, CA, USA, vol. 5021, pp. 148–159 (2003)
7. Tonomura, Y., Akutsu, A., Otsugi, K., Sadakata, T.: VideoMAP and VideoSpaceIcon: tools for automatizing video content. In: Proceedings of ACM INTERCHI '93 Conference, pp. 131–141 (1993)
8. Ueda, H., Miyatake, T., Yoshizawa, S.: IMPACT: an interactive natural-motion-picture dedicated multimedia authoring system. In: Proceedings of ACM CHI '91 Conference, pp. 343–350 (1991)
9. Rui, Y., Huang, T.S., Mehrotra, S.: Exploring video structure beyond the shots. In: Proceedings of IEEE International Conference on Multimedia Computing and Systems (ICMCS), Texas, USA, pp. 237–240 (1998)
10. Pentland, A., Picard, R., Davenport, G., Haase, K.: Video and image semantics: advanced tools for telecommunications. IEEE MultiMed. **1**(2), 73–75 (1994)
11. Sun, Z., Ping, F.: Combination of color and object outline based method in video segmentation. Proc. SPIE Storage Retr. Methods Appl. Multimed. **5307**, 61–69 (2004)
12. Arman, F., Hsu, A., Chiu, M.Y.: Image processing on compressed data for large video databases. In: Proceedings of ACM Multimedia '93, Annaheim, CA, USA, pp. 267–272 (1993)
13. Zhuang, Y., Rui, Y., Huang, T.S., Mehrotra, S.: Key frame extraction using unsupervised clustering. In: Proceedings of ICIP'98, Chicago, USA, vol. 1, pp. 866–870 (1998)
14. Girgensohn, A., Boreczky, J.: Time-constrained keyframe selection technique. Multimed. Tools Appl. **11**, 347–358 (2000)
15. Gong, Y., Liu, X.: Generating optimal video summaries. In: Proceedings of IEEE International Conference on Multimedia and Expo, vol. 3, pp. 1559–1562 (2000)
16. Zhao, L., Qi, W., Li, S.Z., Yang, S.Q., Zhang, H.J.: Keyframe extraction and shot retrieval using nearest feature line (NFL). In: Proceedings of ACM International Workshops on Multimedia Information Retrieval, pp. 217–220 (2000)
17. Hanjalic, A., Lagendijk, R.L., Biemond, J.: A new method for key frame based video content representation. In: Image Databases and Multimedia Search. World Scientific, Singapore (1998)
18. Hoon, S.H., Yoon, K., Kweon, I.: A new technique for shot detection and key frames selection in histogram space. In: Proceedings of the 12th Workshop on Image Processing and Image Understanding, pp. 475–479 (2000)
19. Narasimha, R., Savakis, A., Rao, R.M., De Queiroz, R.: A neural network approach to key frame extraction. In: Proceedings of SPIE-IS & T Electronic Imaging Storage and Retrieval Methods and Applications for Multimedia, vol. 5307, pp. 439–447 (2004)
20. Calic, J., Izquierdo, E.: Efficient key-frame extraction and video analysis. In: Proceedings of IEEE ITCC2002, Multimedia Web Retrieval Section, pp. 28–33 (2002)

21. Liu Tianming, M., Zhang, H.J., Qi, F.H.: A novel video keyframe-extraction algorithm based on perceived motion energy model. IEEE Trans. Circuits Syst. Video Technol. **13**(10), 1006–1013 (2003)
22. Zhang, H.J., Wu, J., Zhong, D., Smoliar, S.W.: An integrated system for contentbased video retrieval and browsing. Pattern Recognit. **30**(4), 643–658 (1997)
23. Fayzullin, M., Subrahmanian, V.S., Picarello, A., Sapino, M.L.: The CPR model for summarizing video. In: Proceedings of the 1st ACM International Workshop on Multimedia Databases, New Orleans, LA, USA, pp. 2–9 (2002)
24. Lagendijk, R.L., Hanjalic, A., Ceccarelli, M.P., Soletic, M., Persoon, E.H.: Visual search in a SMASH system. In: Proceedings of ICIP'96, pp. 671–674 (1995)
25. Ngo, C.-W., Ma, Y.-F., Zhang, H.-J.: Video summarization and scene detection by graph modeling. IEEE Trans. Circuits Syst. Video Technol. **15**(2), 196–305 (2005)
26. Chang, H.S., Sull, S., Lee, S.U.: Efficient Video Indexing Scheme for Content-Based Retrieval. IEEE Trans. Circuits Syst. Video Technol. **9**(8), 1269–1279 (1999)
27. Tieyan, L., Zhang, X., Feng, J., Lo, K.T.: Shot reconstruction degree: a novel criterion for key frame selection. Pattern Recognit. Lett. **25**, 1451–1457 (2004)
28. Fernando, A.C., Canaharajah, C.N., Bull, D.R.: Fade-in and fade-out detection in video sequences using histograms. In: Proceedings of ISCAS 2000—IEEE International Symposium on Circuits and System, vol. IV, pp. 709–712 (2000)
29. Swain, M., Ballard, D.: Color indexing. Int. J. Comput. Vis. **7**(1), 11–32 (1991)
30. Smith, J.R., Chang, S.F.: Tools and techniques for color image retrieval. In: IST/SPIE Storage and Retrieval for Image and Video Databases IV, vol. 2670, pp. 426–437 (1996)
31. Ciocca, G., Gagliardi, I., Schettini, R.: Quicklook^2 : an integrated multimedia system. Int. J. Vis. Lang. Comput. **12**, 81–103 (Special issue on querying multiple data sources) (2001)
32. Gonzalez, R., Woods, R.: Digital image processing. Addison Wesley, Reading, pp. 414–428 (1992)
33. Idris, F., Panchanathan, S.: Storage and retrieval of compressed images using wavelet vector quantization. J. Vis. Lang. Comput. **8**, 289–301 (1997)
34. Scheunders, P., Livens, S., Van de Wouwer, G., Vautrot, P., Van Dyck, D.: Wavelet-based texture analysis. Int. J. Comput. Sci. Inf. Manage. **1**(2), 22–34 (1998)
35. Chetverikov, D., Szabo, Zs.: A simple and efficient algorithm for detection of high curvature points in planar curves. In: Proceedings of the 23rd Workshop of the Austrian Pattern Recognition Group, pp. 175–184 (1999)
36. Latecki, L., DeMenthon, D., Rosenfeld, A.: Extraction of key frames from videos by polygon simplification. In: International Symposium on Signal Processing and its Applications, pp. 643–646 (2001)
37. Lefevre, S., Holler, J., Vincent, N.: A review of real time segmentation of uncompressed video sequences for content-based search and retrieval. Real Time Imaging **9**, 73–98 (2003)
38. Daubechies, I., Sweldens, W.: Factoring wavelet transforms into lifting steps. J. Fourier Anal. Appl. **4**(3), 247–269 (1998)
39. AESS Archive. http://aess.itc.cnr.it/index.htm
40. MAIS Consortium, Mais: Multichannel Adaptive Information Systems. http://black.elet.polimi.it/mais/

Gianluigi Ciocca received his degree (Laurea) in Computer Science at the University of Milan in 1998, and since then he has been a fellow at the Institute of Multimedia Information Technologies and at the Imaging and Vision Laboratory in the ITC Institute of the Italian National Research Council, where his research has focused on the development of systems for the management of image and video databases and the development of new methodologies and algorithms for automatic indexing. He is currently a PhD student in computer science at the Department of Information Science, Systems Theory, and Communication (DISCo) of the University of Milano-Bicocca, working on video analysis and video abstraction.

Raimondo Schettini is an associate professor at DISCo, University of Milano Bicocca, where he is in charge of the Imaging and Vision Lab. He has been associated with Italian National Research Council (CNR) since 1987. He has been team leader in several research projects and published more than 160 refereed papers on image processing, analysis and reproduction, and on image content-based indexing and retrieval. He is an associated editor of the Pattern Recognition Journal. He was a co-guest editor of three special issues about Internet Imaging (Journal of Electronic Imaging, 2002), Color Image Processing and Analysis (Pattern Recognition Letters, 2003) and Color for Image Indexing and Retrieval (Computer Vision and Image Understanding, 2004). He was General Co-Chairman of the 1st Workshop on Image and Video Content-based Retrieval (1998), the First European Conference on Color in Graphics, Imaging and Vision (2002) and the EI Internet Imaging Conferences (2000–2006).