ABSTRACT
We propose an innovative, general purpose, approach to the selection and hierarchical representation of key frames of a video sequence for video summarization. In the first stage the shot detection module performs the video structural analysis; in the second stage the key frame extraction module creates the visual summary; and the last stage the summary post-processing module, after a pre-a classification aimed remove meaningless key frames, create a multilevel storyboard that the user may browse.

Categories and Subject Descriptors
H.3.1 [Information storage and retrieval]: Content Analysis and Indexing – Abstracting methods, Indexing methods; I.4.10 [Image Processing and Computer Vision]: Image Representation – Multidimensional; I.5.3 [Pattern Recognition]: Applications – Computer Vision.

General Terms
Algorithms

Keywords
Dynamic video summaries, summary post-processing, supervised and unsupervised classification, multilevel visual summaries.

1. INTRODUCTION
Very large public and personal 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 is simply redundant, and we must find a way to retain only the information strictly needed for functional access, browsing and querying. Representation of video contents using a set of frames (key frames) is one of the most common techniques for video summarization. Summaries composed of key frames can resume the video contents in a rapid and compact way: users can grasp the overall contents more quickly from key frames than by watching a set of video sequences. Besides providing video browsing capability and content description, key frames act as video “bookmarks” that designate the interesting events captured, supplying direct access to video subsequences.

Consumer’s Personal Video Recorders (PVRs) permit the storage of a large amount of video data and allow the users to perform different tasks such as navigate the different recorded programs, non-linear access to the video content and limited video editing. Automatically extracted key frames would provide a fast and easy way to manage, browse and construct a visual table of content of the recorded videos [7][13]. Similar applications can be devised for the design of software to be used with professional and personal videos acquired with digital video cameras. We propose an innovative approach to the selection and hierarchical representation of key frames of a video sequence for video summarization. The dynamic algorithm for key-frame identification escapes the complexity of existing methods: key frame are simply extracted on the fly by detecting curvature points within the curve of the cumulative frame differences. The visual summary so obtained, exploiting only low level features, may contain uninteresting and meaningless key frames (overexposed and underexposed key frames; close-ups with very few details, etc). Moreover, successive very similar key frames can be extracted disturbing the visual summary since they convey the same information. Finally, the summaries may be formed by hundreds or even thousands of key frames and are therefore dispersive since they require intensive browsing. The hierarchical representation strategy presented in this paper, exploiting both low level and high level features, makes it possible to create exhaustive and not redundant visual video summaries. Three steps compose the algorithm: the first removes the meaningless key frames; the second groups the key frames into both visually and semantically homogenous clusters to allow hierarchical summary presentation; and the third identifies the default summary level. Staring from these reduced set of key frames, the users can browse the video at different level of summarization. Experiments over 14 videos of about 4 hours duration shows the effectiveness of the proposed approach.

2. STORYBOARDS GENERATION
A general purpose video analysis pipeline with particular attention to the generation of dynamic visual summaries in form of storyboards is presented. The pipeline, shown in Figure 1, uses innovative solutions for the most important tasks of video analysis and is composed of three modules: the shot detection module which performs the video structural analysis; the key frame extraction module which creates the visual summary; and the summary post-processing module which creates a multilevel summary.
to a decision module which gives the final response. This allows a version of the algorithm proposed by Fernando et al. [8].

The use of following assorted visual descriptors provides a more precise representation of the frame and captures slight variations between the frames in a shot: 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 following assorted visual descriptors provides a more precise representation of the frame and captures slight variations between the frames in a shot:

**Color histogram.** This is frequently used to compare images because they are simple to compute, and tend to be robust regarding small changes in camera viewpoint. The color histogram we use is composed of 64 bins determined by sampling groups of meaningful colors in the HSV color space [4]. The use of the HSV color space allows us to carefully define groups of colors in terms of Hue, Saturation and Lightness.

**Edge direction histogram.** It is composed of 72 bins corresponding to intervals of 2.5 degrees. Two Sobel filters are applied to obtain the gradient of the horizontal and the vertical edges of the luminance frame image [10]. 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 the 4% of the gradient maximum value in order to remove from the histogram computation edges derived from background noise.

**Multiresolution wavelet statistics.** This provides representations of image data in which both spatial and frequency information are present [11]. 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 a 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. In our procedure the features are extracted from the luminance image using a three-step Daubechies multiresolution wavelet decomposition that uses 16 coefficients and producing ten sub-bands [17]. Two energy features, the mean and standard deviation of the coefficients, are then computed for each of the 10 sub-band 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. The distance between two color histograms \(d_{CH}\) uses the intersection measure. The difference between two edge direction histograms \(d_{ED}\) is computed using the Euclidean distance as such in the case of two wavelet statistics \(d_{WT}\). The three resulting values are mapped into the range \([0,1]\) and then combined to form the final frame difference measure \(d_{FMC}\) as follow:

\[
d_{FMC} = (d_{CH} \cdot d_{WT}) + (d_{CH} \cdot d_{TH}) + (d_{TH} \cdot d_{WT})
\]  

(1)

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 contents. 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 features can overcome these issues but pose the problem of their combination. In content-based retrieval systems, the features are combined by weighting them with suitable factors which are usually task-dependent [4]. 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 \(d_{FMC}\) values only if at least two of the single differences exhibit high values (and thus two of the visual attributes emphasize the frame dissimilarity). The identification of cuts (and flashes) in the sequence of frame differences can be

---

**Figure 1. The storyboards generation.**
summarized as follows: let \( d^* = d_{HWD}(t, t-1) \) be the frame difference under examination.

1. If it is below the threshold \( T \) then examine the next difference, otherwise go to step 2.
2. Checking for isolated peaks.
   If all the differences \( d_{HWD}(x, x-1) \) with \( x \in [t-w, t-1] \) and \( x \in [t+1, t+w] \) are all below a lower threshold \( T_x \) then mark the frame at position \( t \) as a possible cut. Otherwise get the next difference, and go to step 1.
3. Checking for flashes.
   If the difference \( d_{HWD}(t+1, t) \) is above the threshold \( T \) and the difference \( d_{HWD}(t, t-1) \) is below the threshold \( T \), then there is a flash at position \( t \), get the next difference, and go to step 1. Otherwise go to step 4.
4. Declare a cut at position \( t \).

### 2.1.2 Fade Detection Algorithm

As stated before, the fade detection algorithm is based on the approach proposed by Fernando et. al [8]. The modifications aim to cope with some drawbacks of the original algorithm and in particular with its sensitivity to the length of the fade sequences and less-than-ideal fade transitions. The algorithm can be summarized as follows:

1. Smooth frame at time \( t \), \( F(t) \), with a low-pass filter.
2. Compute the pixel intensity standard deviation from the central frame's region. The central region is obtained by eliminating the most external pixels.
3. Given \( n \) successive frames \( F(t-n+1), F(t-n+2), \ldots, F(t) \), and the corresponding computed standard deviation values \( \sigma(t-n+1), \sigma(t-n+2), \ldots, \sigma(t) \)
   3.1 A fade out sequence is identified at position \( j \) with \( t-n+1 \leq j < t \) if the following conditions hold:
      a) \( \sigma(j) > T \) and \( \sigma(j+1) < T \), and
      b) \( \sigma(j) > \sigma(j+1) \) and \( \sigma(j) - \sigma(j+1) < kT \)
      for every \( t-n+1 \leq j < t-1 \)
   3.2 A fade in is identified at position \( j \) with \( t-n+1 \leq j < t \) if the following conditions hold:
      a) \( \sigma(j) < T \) and \( \sigma(j+1) > T \), and
      b) \( \sigma(j) < \sigma(j+1) \) and \( \sigma(j) - \sigma(j+1) > kT \)
      for every \( t-n+1 \leq j < t-1 \)
   3.3 If both a fade in and a fade out are identified, ignore the results.
4. Increase \( t \) and go to 1.

The cut and fade detection can be performed on-the-fly by implementing a data buffer of size \( n \) with FIFO policy. Every time a new frame difference or standard deviation is computed, it can be inserted in the buffer and then the values within it analyzed. We design the shot detection algorithm to be a general purpose one aimed to be used with any kind of video. In fact, it relies only on the visual information of the frames to detect the different editing effects within a video sequence. In case the source of a video sequence is a digital camcorder, to determine the shots the metadata automatically stored in the sequence can be used instead.

### 2.2 Key Frame Extraction

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 of only. The proposed key frame extraction algorithm functions as shown in Figure 2.

![Key Frame Extraction](image)

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. These changes are identified exploiting the same frame difference measure \( d_{HWD} \) used in the shot boundary detection in Section 2.1. The frame difference values 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: 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 mid points between each pair of consecutive curvature points. To detect the high curvature points we use the algorithm proposed by Chetverikov et al. [3].

![High Curvature Points](image)

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 of $P$. 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 Figure 3. Defining the distance between point $s$ and $O$ as $d_{sO}$, the distance between points $P$ and $R$ as $d_{PR}$, and the distance between points $O$ and $P$ as $d_{OP}$, the opening angle $\alpha$ corresponding to the triangle $OPR$ is computed as:

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

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

$$d_{\min} \leq |P_x - O_x| \leq d_{\max}$$

$$d_{\min} \leq |P_x - R_x| \leq d_{\max}$$

and the constraint on the angle values

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

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 $P$:

$$\alpha(P) = \min \left\{ \alpha = \arccos \frac{d_{OP}^2 + d_{PR}^2 - d_{OR}^2}{2 \cdot d_{OP} \cdot d_{PR}} \right\}$$

If a point has no admissible triangles, the point is rejected assigning it an angle default value of $\pi$. 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 $N$, that is if:

$$\alpha(P) > \alpha(N)$$

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

$$|P_s - N_s| \leq d_{\max}$$

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 midpoint between two consecutive high curvature points. 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. 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. 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. Since 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.

A candidate point $P$ is discarded if it has a sharper valid neighbor $N$, that is if:

$$\alpha(P) > \alpha(N)$$

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

$$|P_s - N_s| \leq d_{\max}$$

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 midpoint between two consecutive high curvature points. 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. 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. 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. Since 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 the case of fading or dissolving effects, the key frames extraction algorithm is not able to detect them, their visual evolution does not present sharp changes and they will not have corner points 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. Figure 4 shows and example of a visual summary extracted from a documentary. The original video is a MPEG-4 sequence 30 minutes and 30 seconds long, composed of 45,753 frames (384x288 pixels at 25fps). The video is an excerpt from a documentary on the “Summer White Carnival” at Cegni, showing a feast in progress, and a participant preparing for the traditional Carnival masquerade.

2.3 Summary Post-Processing

The summary post-processing module is composed by three different steps (Figure 5). The first removes meaningless key frames, using supervised classification performed by a neural network on the basis of pictorial features derived directly from the frames, together with others derived from the processing of the frames by a visual attention model algorithm. The second step provides for the grouping of the key frames into clusters to allow multilevel summary using both low level and high level features. The third step identifies the default summary level that is shown to the users: starting from this set of key frames, the users can then browse the video content at different level of detail.

To remove meaningless key frames, we extract a set of features that describe them in terms of quality and information supplied. To assess image quality, we use some of the features considered in judging the quality of images taken by digital cameras: the percentage of dark and bright pixels, to identifying overexposed and underexposed images; the dynamic range of the image, to single out flat-looking images; and the color balance [19]. To assess the amount of information the key frames convey (that is, their “informativeness”), we use a visual attention model proposed by Corchs et. al [5] to locate the Regions of Interest (ROIs) on a saliency map. This visual attention model can detect those portions of the input image where highly informative contents are located, and suppress the remaining parts. It produces a map of activities of the original image: high values in the resulting map indicate areas of high neural activity, i.e. areas where we expect visual attention to be focused, and thus significant information to be located. Figure 6 shows the processing steps followed to obtain the ROIs of a given image. The first step is the computation of the neuronal activity map by the visual attention models. The second is the binarization of the attention map in order to retain only those regions of greater neural activity. The values of the attention map are analyzed, and the statistics regarding their distribution are used to determine a threshold. Values above the computed threshold are considered to belong to the region of interest. The regions not selected are assumed to concern non-relevant and noisy data. We choose to use as threshold the average of the activity values. Only those pixels for which the corresponding activity is above the average are retained.

The result of the binarization process is a preliminary binary mask representing the possible location of the ROIs. Since the binary mask may be composed of isolated pixels or fragmented regions, the third step in the processing pipeline is binary morphological filtering, applied to remove noisy pixels, and obtain regions with smooth borders and uniform areas. The result of the third step is a mask image containing the locations of all the filtered ROIs. Figure 7 shows some examples of maps of activities and the ROIs extracted from uninformative images (7a and 7b), and informative images (7c and 7d).

Figure 5. Pipeline of the proposed post-processing algorithm.

2.3.1 Removal of Meaningless Key Frames

To remove meaningless key frames, we extract a set of features that describe them in terms of quality and information supplied. To assess image quality, we use some of the features considered in judging the quality of images taken by digital cameras: the percentage of dark and bright pixels, to identifying overexposed and underexposed images; the dynamic range of the image, to single out flat-looking images; and the color balance [19]. To assess the amount of information the key frames convey (that is, their “informativeness”), we use a visual attention model proposed by Corchs et. al [5] to locate the Regions of Interest (ROIs) on a saliency map. This visual attention model can detect those portions of the input image where highly informative contents are located, and suppress the remaining parts. It produces

Figure 6. ROIs extraction pipeline.

Figure 7. Examples of ROIs extraction. Original frame (left), the corresponding maps of activities (center), and regions-of-interest extracted (right). a1 and b1 represent uninformative images while c1 and d1 represent informative images.
Intuitively, images with a large volume of ROIs are more informative than images with a small volume of ROIs. The location of these ROIs is also helpful in discriminating between informative and uninformative images. If the ROIs are located near the center of the image, the object of attention is clearly in the foreground. If, instead, the ROIs are located near the borders of the image, the object of interest is in the background, and not clearly identifiable. Consequently two features extracted as criteria of “Informativeness” are the percentage of pixels belonging to ROIs within a central region of the image, and their dispersion with respect to the center of the image. The central region has been set at about 65% of the whole image. In deciding whether or not to reject a key frame, candidate frames are classified in two groups: key frames to reject, and key frames to retain. Due to the small number of features involved (eight in all here), the classification of the key frames is performed using a neural network classifier [15]. For our purposes, the neural network was composed of one input node for each feature value (i.e. eight input nodes) plus a node with a bias term; a number of hidden layers, each with the same number of nodes as the input (we experimented with one, two, and three hidden layers), and a single final node which gave an output value in the range of $[0,..,1]$. The output could be interpreted as the probability that the processed key frame should be discarded. Figure 8 shows the summary of Figure 4 after the key frames removal stage. As it can be seen, nine key frames have been removed: among the others, the two color bars (frames 217, and 45262), some strong close ups (frames 35890, 36002, 36069, and 38962), and a frame mainly showing the shadow of a person leaving the room (31719).

### 2.3.2 Key Frames Grouping

The key frame grouping algorithm that we propose is conceptually different from the scene clustering task where key similar key frames that are spread along the summary are deemed to belong to the same scene, and grouped together. We are mainly interested in merging similar key frames while preserving the temporal ordering of the remaining key frames regardless of the shot boundaries. Key frames belonging to the same shot may be merged in different clusters if they are sufficiently dissimilar in appearance and content. Instead of using only simple pictorial features to describe the key frame content (as most of the scene clustering algorithms proposed in the literature do), we use a two level description approach. The key frames are described by low level and high level (semantic) features. The low level features are the same used in Section 2.1.1. Given this set of features, the pictorial signature ($Ps$) of a key frame $KF$ can be defined as:

$$Ps(KF) = (H, W, D)$$  \hspace{1cm} (8)

where $H$ represents the color histogram, $W$ the wavelet statistics and $D$ the edge direction histogram. Two pictorial signatures are evaluated using, again, the $d_{HWD}$ difference measure defined previously (Equation 1):

$$D_{HWD}(Ps_1, Ps_2) = d_{HWD}(Ps_1, Ps_2)$$  \hspace{1cm} (9)

The high level features are obtained by applying the classification strategies described in [16], where the images are classified as indoor, outdoor, or close-up images. The classification is based on the use of ensembles of decision trees, often called decision forests.

The trees of the forests are constructed according to CART methodology [2]. The features used are related to color (moments of inertia of the color channels in the HSV color space, and skin color distribution), texture and edge (statistics on wavelets decomposition and on edge and texture distributions), and composition of the image (in terms of fragmentation and symmetry). To fully exploit the fact that trees allow a powerful
use of high dimensionality and conditional information, all the features are taken together, letting the training process perform complexity reduction, and redundancy detection. An ambiguity rejection option is also included, for a more accurate classification. The final decision to classify an image in one of the three classes, or reject it, is made by a majority vote among the decision results of the trees in the forest. For a more detailed description of the classification process, see [16]. Examples of key frame classification results are presented in Figure 9. Instead of assigning an image to a single class we have used the probabilities that a given image belongs to each class as a semantic histogram signature. The semantic signature (Ss) of a key frame KF is then defined as:

\[
Ss(KF) = (I, O, C)
\]

(10)

where I, O and C are the percentage of the frame belonging to the Indoor, Outdoor and Close-up classes respectively. Two semantic signatures are compared using the Euclidean distance:

\[
\text{Diff}_{\text{loc}}(S_{x1}, S_{x2}) = \sqrt{(I_{1} - I_{2})^2 + (O_{1} - O_{2})^2 + (C_{1} - C_{2})^2}
\]

(11)

The overall difference (Diff) between two key frames is computed by a linear combination of the pictorial signature difference and the semantic signature difference (to simplify the notation, we omit the arguments of the differences):

\[
\text{Diff}(KF_1, KF_2) = \alpha(\text{Diff}_{\text{pict}}) + (1 - \alpha)\text{Diff}_{\text{loc}}
\]

(12)

The factor \(\alpha\) is used to weight the contribution of one signature over the other. In our experiments the weight was set at 0.5, equally weighting the two signatures. The use of features at different levels of abstraction allows us to reduce the error made in comparing two key frames. If both the pictorial content and the semantic content are similar the two key frames probably belong to the same shooting sequence; they are consequently merged together. To group similar key frames a hierarchical clustering based on the complete link strategy has been adopted adding few together. To group similar key frames a hierarchical clustering algorithm is used. The clusters are composed of representative key frames (representative element) that represents the visual summary at different levels of detail (i.e. with different numbers of clusters/key frames). This allows the user to inspect the contents of the video by navigating throughout the levels. The problem of the multilevel summary is to decide which summary to present to the user as the optimal or default summary level. When the number of levels is low (i.e. the number of key frames composing the original visual summary is low), the user can easily browse all the levels until he eventually finds the one with the information he is searching for. When the original summary is composed of hundreds of key frames, browsing through all the levels is cumbersome and time consuming. It is necessary to define a strategy to select, if it exists, the summary that is least redundant in terms of pictorial information (i.e. few similar key frames). It also affords an optimal starting point for browsing the remaining summary levels.

The idea underlying the strategy for detecting the default summary level is that the frame differences used in merging the clusters in the previous section be considered merging costs, that is, the costs of reducing the summary by one key frame. In the initial phase of the clustering process, the costs will be relatively small, that is, the merged clusters will have small merging costs because the representative key frames are similar, and their merging will not significantly reduce the summary information contents. As clustering continues, the merging costs will increase, meaning that the representative key frames will be increasingly dissimilar, and their merging will result in a summary with fewer information contents. We select the level at which the merging costs rise significantly as the level of the default summary. Starting from there and moving toward lower levels we obtain summaries with more key frames and redundant information. Moving toward higher levels, we have summaries with few key frames and more compact contents. To select the clustering level we have used an approach based on the peer-group filtering (PGF) scheme proposed by Deng et al [6][12]. The authors employed it to filter image noise and to quantize color space for the purpose of image enhancement. In [14] the PGF scheme has been employed to detect dominant scenes in sports videos. The objective of the PGF is to group a set of data into two classes by minimizing intra-
The level corresponding to the maximum value $D$, indicates the partition (in terms of Fisher’s analysis) where the costs assigned to each group are homogeneous, and the costs of the second group are visibly higher than those of the first. Reducing the summary by one key frame at a time from this point on, means merging pairs of dissimilar key frames (greater differences), with a consequent loss of information. The default clustering level for the summary in Figure 8 is shown in Figure 10. Image a, is the plot of the Fisher’s indexes with a maximum in the 27th position. Image b shows the summary corresponding to the 27th clustering level. If the maximum value is very close to the first or last clustering level, one of the two groups will be composed of very few elements and the Fisher analysis has not been able to identify a clear separation between the costs. We may conclude that, for that summary, the default level does not differ substantially from that of the original summary at level 0 which can then be used as the default level.

3. FINAL REMARKS

We propose an innovative, general purpose, approach to the selection and hierarchical representation of key frames of a video sequence for video summarization. The multilevel visual summaries created are used to offer to the users a compact presentation of the video contents, a meaning for browsing its contents at different levels of granularity, and direct links to the subsequences that compose the video. Users can then quickly browse through a video sequence, rapidly get an overview of the contents, and navigate to different levels of detail to locate the segments of interest. However, since different users may be interested in different levels of detail, it is also necessary to provide a hierarchical structure containing different views of video contents, from a coarse view showing the overall plot, to finer views containing an increasingly number of details. Since all the summary’s post processing steps (removal of meaningless key frames and classification) rely on semantic information which no objective quality measure can effectively incorporate, the post-processing algorithm was heuristically tested by domain experts on a set of videos belonging to the "Archivio di Etnografia e Storia Sociale - AESS" [18]. These experts manage video footages on a daily basis, manually extracting relevant information the videos to use for content cataloging and publication through distributed channels. The test set was composed of 14 non-professional videos, about 4 hours of footage. The experts evaluated the processed summary in terms of compactness, and information contents as well as the effectiveness of the multilevel summary. As the post-processing algorithm does not use previous knowledge about the video contents, nor is any assumption made about the input data, it can be used in different domains as a general purpose algorithm. Nevertheless, some improvements can be made. The key frame removal stage could be extended with more pictorial quality features (both low level and high level) in order to better cover the many factors that can cause a user to reject a frame (e.g. wrong skin tone, half faces, etc...). The key frame grouping stage could also be extended. We have introduced three generic classes for the classification of the key frames but more classes can be added in the decision trees to enlarge the semantic dictionary. The generic approaches used in the whole post processing pipeline, mean that it can easily be specialized to support domain specific applications by taking into account the appropriate pictorial and semantic properties.
4. REFERENCES


