An Integrated Scheme for Automated Video Abstraction Based on Unsupervised Cluster-Validity Analysis

Alan Hanjalic and HongJiang Zhang, *Senior Member, IEEE*

***Abstract*—*Key frames* and *previews* are two forms of a video abstract, widely used for various applications in video browsing and retrieval systems. We propose in this paper a novel method for generating these two abstract forms for an arbitrary video sequence. The underlying principle of the proposed method is the removal of the visual-content redundancy among video frames. This is done by first applying multiple partitional clustering to all frames of a video sequence and then selecting the most suitable clustering option(s) using an unsupervised procedure for cluster-validity analysis. In the last step, key frames are selected as centroids of obtained optimal clusters. Video shots, to which key frames belong, are concatenated to form the preview sequence.**

***Index Terms*— Clustering, cluster-validity analysis, content-based video retrieval, content classification, video content analysis.**

I. Introduction

A *video abstract* is a compact representation of a video sequence and is useful for various video applications. For instance, it provides a quick overview of the video-data-base content and enables fast access to shots, episodes, and entire programs in video browsing and retrieval systems. There are two basic forms of a video abstract:

- a *preview* sequence, being the concatenation of a limited number of selected video *segments (key video segments)*;
- a set of *key frames,* being a collection of suitably chosen frames of a video.

A preview sequence is made with the objective of reducing a long video into a short sequence that is often used to help the user to determine if a video program is worth viewing in its entirety. It either provides an impression about the entire video content or contains only the most interesting video segments. We distinguish between these two types of previews and define them as the *summary sequences* and *highlights,* respectively.

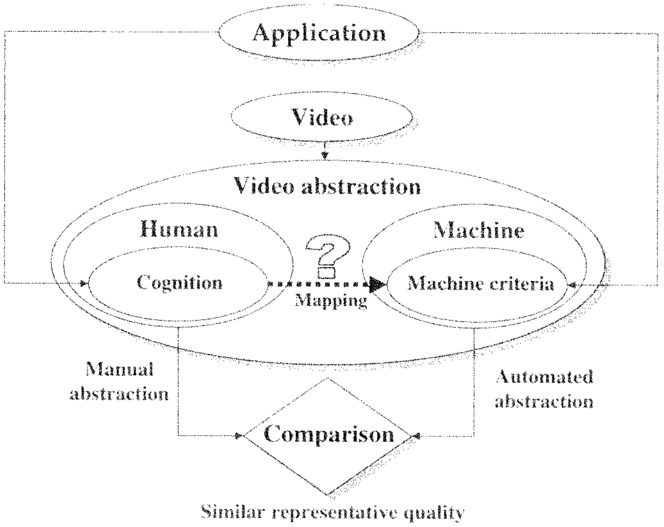


Fig. 1. Manual versus automated video abstraction.

Key frames are most suitable for content-based video browsing, where they can be used to guide a user to locate specific video segments of interest. Furthermore, key frames are also effective in representing visual content of a video sequence for retrieval purposes: video indexes may be constructed based on visual features of key frames, and queries may be directed at key frames using image retrieval techniques [24]. Also, shot comparisons involved in some high-level video processing and analysis steps can benefit from visual features captured in key frames [10], [21]. Similarly, as in the case of preview sequences, key frames can also be extracted in two different ways, either to capture the most memorable scenes of a video or to just summarize the entire video in terms of its visual content. Consequently, here we also use the terms such as *highlighting* and *summarizing* key frames.

As illustrated in Fig. 1, a video can be abstracted either manually or automatically. If key frames and key video segments are extracted manually, they will comply to human cognition. That is, these key frames/segments will be selected based on human understanding of the content of a video and human perception of representativeness and quality of a frame or a segment. For instance, key frames or key video segments can be extracted here based on the role that the persons and objects captured therein play in the context of the target application. One can choose the most representative ones (e.g., taken under the best camera angle, best revealing the interesting action) from many candidate frames or segments. Furthermore, it is expected that no blurred frames or "dark" segments are extracted, nor those with coding artifacts, interlacing effects, etc.

Manuscript received October 31, 1998; revised August 25, 1999. This paper was recommended by Guest Editor S.-F. Chang.

A. Hanjalic is with the Delft University of Technology, 2628 CD Delft, The Netherlands.

H. Zhang is with Microsoft Research, Beijing, China.

Publisher Item Identifier S 1051-8215(99)09585-3.

Reducing human involvement in the video abstraction process by developing fully automated video analysis and processing tools steadily gains more importance from both the production and the user end of video programs. On the one hand, a continuously growing video production and increasing number of different services offered to customers require enormous manpower at the production end for processing and preparing the videos for their distribution and placement on the market. On the other hand, if video abstraction is to be performed at the user end, for instance, in emerging digital storage systems containing some data-processing power [18], a full automation of such systems is crucial, since users at home want to be entertained and not burdened with programming or adjusting their video equipment.

While the need for automating the video-abstraction procedure is strong, the possibilities for its practical realization are limited. The first limitation is related to the fact that it is highly difficult to develop a system capable of automatically capturing the highlights of a video. This is mainly due to the fact that defining which video segments are the highlights is very subjective process, and thus it is difficult to obtain objective ground truth. Furthermore, we are still missing the feasibility to efficiently map human cognition into the automated abstraction process such that similar abstraction results are generated manually and automatically.

In this paper, we present a method for automatically producing an abstract of an arbitrary video sequence. The method is based on cluster-validity analysis and is designed to work without any human supervision. The produced video abstract consists of a set of key frames and a preview sequence. Since we were aware of the above limitations, we followed the objective of *summarizing* the content of a given video sequence, rather than finding its highlights. The role of subjectivity in the video summarization process is significantly reduced, which can be explained by the fact that a video summary ideally contains *all* relevant elements of the video content (faces, objects, landscapes, situations, etc.) and not a subjective selection of these elements, such as highlights. In view of the second limitation, the abstracting method presented in this paper does not attempt to provide an abstract containing precisely the same key frames or segments as the one formed manually. Its objective is rather to summarize the sequence in a way similar to the manual one, in terms of the video material captured by the abstract and the obtained abstract size. This can be explained with an example of a simple dialog-like sequence, illustrated in Fig. 2, consisting of interchanging shots showing each of the two content components A and B. Since a person would summarize such a sequence by taking only two key frames/segments, one for each of the content components, the same needs to be obtained automatically using our method, although the chosen key frames and key segments can be taken at different time instances.

In Section II, we first review some of the representative previous approaches to video abstraction, after which we present our abstraction method in Section III. There, first, the procedures for clustering and cluster-validity analysis are explained. This is followed by a description of how the abstract is formed on the basis of clustering results and by defining the application scope of the proposed method. Section IV shows the experimental evaluation, and Section V concludes this paper.

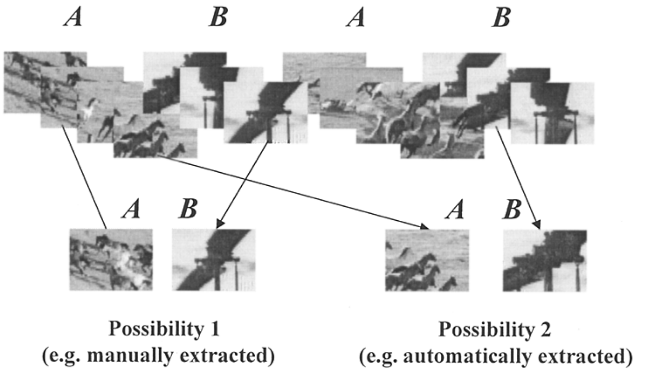


Fig. 2. Two of several possible key-frame sets containing different frames but capturing the same visual material.

II. Previous Work on Video Abstraction

A. Key-Frame Extraction

Automated extraction of key frames has been addressed by many researchers [1], [2], [4], [6]–[9], [13], [16], [17], [19]–[26]. A first attempt to automate the key-frame extraction was to choose as a key frame the frame appearing after each detected shot boundary [16]. However, while being sufficient for stationary shots, one key frame does not provide an acceptable representation of the visual content in dynamic sequences. Therefore, methods were needed to extract key frames compliant to visual content variations along a video sequence. One of the first key-frame extraction approaches developed in view of this objective is presented in [22], with all details given in [25]. There, key frames are extracted in a sequential fashion for each shot separately. The first frame of a shot is always chosen as a key frame. Then, similar methodology is applied as for detecting shot boundaries. The difference $z(F_{\mathrm{last}}, k)$ is computed between the current frame k of a sequence and the last extracted key frame F_{last} using color histograms. If this difference exceeds a given threshold T, the current frame is selected as a new key frame, that is

$$\begin{aligned} &\mathit{Step1}\colon \quad F_{\mathrm{last}} = 1 \\ &\mathit{Step2}\colon \quad \forall k \in [2, S] \text{ if } z(F_{last}, k) > T \Rightarrow F_{\mathrm{last}} = k. \end{aligned} \tag{1}$$

Here, S is the number of frames within a shot. The extraction procedure (1) is then adapted by using the information on dominant or global motion resulting from camera operations and large moving objects, according to a set of rules. For a zooming-like shot, at least two frames will be extracted, at the beginning and at the end of a zoom. The first one represents a global and the other one a more detailed view of a scene. In cases of panning, tilting, and tracking, the number of frames to be selected will depend on the rate of visual-content variations: ideally, the visual content covered by each key frame should have little overlap, or each frame should capture a different object activities. Usually frames that have less than 30% overlap in their visual content are selected as key frames. A key-frame extraction method similar to (1) can also be found in [21], though without usage of the motion information.

In the approach presented in [8], the authors first compute the discontinuity value between the current frame k and the N previous frames. This is done by comparing the color histogram of the frame k and the average color histogram of the previous N frames, that is

$$z(k,\{k-1,\ldots,k-N\}) = \sum_{j=k-1}^{k-N}\sum_{e=Y,U,V}\left|H_k^e(i)-\frac{1}{N}\sum_{j=k-1}^{k-N}H_j^e(i)\right|. \quad (2)$$

If the discontinuity value (2) exceeds the prespecified threshold T, the current frame k is extracted as a new key frame F_{last}, i.e.,

$$\text{if } z(k,\{k-1,\ldots,k-N\}) > T \Rightarrow F_{\text{last}} = k. \quad (3)$$

A possible problem with the extraction methods described above is that the first frame of a shot is always chosen as a key frame, as well as those frames lying in shot segments with a varying visual content. As discussed in [7], when choosing a frame lying close to the beginning or to the end of a shot, there is a probability for that frame of being a part of a dissolve effect at the shot boundary, which strongly reduces its representative quality. The same can be said for frames belonging to shot segments of high camera or object motion (e.g., strong panning or a zoomed object moving close to the camera and covering most of the frame surface). Such frames may be blurred, and thus in some cases not suitable for extraction.

A solution to this problem can be found in [4], where the authors first represent a video sequence as a curve in a high-dimensional feature space. A 13-dimensional feature space is formed by the time coordinate and three coordinates of the largest "blobs" (image regions) using four intervals (bins) for each luminance and chrominance channel. Then the authors simplify the curve by using the multidimensional curve splitting algorithm. The result is, basically, a linearized curve, characterized by "perceptually significant" points, which are connected by straight lines. A key-frame set of a sequence is finally obtained by collecting frames found at perceptually significant points. With a splitting condition that checks the dimensionality of the curve segment being split, the curve can be recursively simplified at different levels of detail, i.e., with different densities of perceptually significant points. The final level of detail depends on the prespecified threshold, which evaluates the distance between the curve and its linear approximation. A potential major problem of this approach is the difficulty to evaluate the applicability of obtained key frames, since there was no comprehensive user study to prove that the extracted key frames lying at "perceptually significant points" capture all important instances of a video, or that there is a clear connection between perceptually significant points and most memorable key frames (highlights).

A different type of key-frame extraction approach is proposed in [26]. There, all frames in a video shot are classified into M clusters, where this final number of clusters is determined by a prespecified threshold T. A new frame is assigned to an existing cluster if it is similar enough to the centroid of that cluster. The similarity between the current frame k and a cluster centroid c is computed as the intersection of two-dimensional hue–saturation (HS) histograms of the hue–saturation–value (HSV) color space. If the computed similarity is lower than the prespecified threshold T, a new cluster is formed around the current frame k. In addition, only those clusters that are larger than the average cluster size in a shot are considered as key clusters, and the frame closest to the centroid of a key cluster is extracted as a key frame.

Extraction of key frames in all approaches discussed above is based on threshold specification. The thresholds used in [4], [22], and [26] are heuristic, while the authors in [8] work with a threshold obtained by using the technique of Otsu [15]. By adjusting the threshold, the total number of extracted key frames can be regulated. However, such regulation can be performed only in a global sense, meaning that a lower threshold will lead to more key frames, and vice versa. The exact or at least approximate control of the total number of extracted key frames is not possible. First, it is difficult to relate certain threshold value to the number of extracted key frames. Second, if the same threshold value is applied, it can lead to a different number of extracted key frames for different sequences.

A practical solution for this problem is to make the threshold directly related to the extraction performance. An example is the threshold specification in form of the maximum tolerable number of key frames for a given sequence. An approach using this sort of thresholds can be found in [17]. There, two thresholds need to be prespecified: r, controlling which frames will be included in the set, and N, being the maximum tolerable number of key frames for a sequence. Key-frame extraction is performed by means of an iterative partitional-clustering procedure. In the first iteration step, a video sequence is divided into consecutive clusters of the same length L. The difference is computed between the first and the last frame in each cluster. If the difference exceeds the threshold r, all frames of a cluster are taken as key frames. Otherwise, only the first and the last frame of the cluster are taken as key frames. If the total number of extracted frames is equal to or smaller than the tolerable maximum N, the extraction procedure is stopped. If not, a new sequence is composed out of all extracted frames and the same extraction procedure is applied. The biggest disadvantage of this method is the difficulty of specifying the threshold r, since it is not possible to relate the quality of the obtained key-frame set to any specific r value.

A better alternative method to [17] was proposed in [9], which does not require any other threshold value but the maximum allowed number of key frames for a given video. There are two steps in this approach. First, the assignment of a number of key frames for each shot is carried out based on the content variation of a shot *and* that of the entire sequence. The content variation of a given sequence is defined as the sum of all frame-to-frame differences measured along the entire sequence. The key-frame assignment is done such that the sum of all assigned key frames along the sequence is close to a given maximal number of allowable key frames N for the entire sequence. The number N can be adjusted if we know *a priori* the type of the program to be processed. The assignment step is followed by a threshold-free and objective procedure to optimally distribute the assigned number of key frames in each video shot. The optimal distribution of key frames in each shot is performed at this second step using a *numerical algorithm* to optimize the representation of the distribution with respect to a given measure of the content flow dynamics along each shot. However, predefining the absolute number of key frames without knowing the video content may be problematic in some cases: when assigning two key frames for a talking head sequence of 30 min, one of them can be considered as redundant. In addition, assigning the same number of key frames to, for instance, two video sequences of the same length does not guarantee the same level of visual abstraction since the contents of the two sequences may have different levels of abstraction and/or totally different levels of activities.

If the total number of extracted key frames is regulated by a threshold, the qualities of the resulting key-frame set and of the set obtained for the same sequence but based on human cognition are not necessarily comparable. For instance, if the threshold is too low, too many key frames are extracted, characterized by a high redundancy of their visual content. By a high threshold, the resulting key-frame set might be too sparse. Especially if the rate of visual-content change allows for only one optimal set of key frames for the best video representation, finding the threshold value providing such a key-frame set is a highly difficult task.

Authors in [2] and [19] aim at avoiding this problem and propose threshold-free methods for extracting key frames. In [2], the temporal behavior of a suitable feature vector is followed along a sequence of frames, and a key frame is extracted at each place of the curve, where the magnitude of its second derivative reaches the local maximum. A similar approach is presented in [19], where local minima of motion are found. First, the optical flow is computed for each frame, and then a simple motion metric is used to evaluate the changes in the optical flow along the sequence. Key frames are then found at places where the metric as a function of time has its local minima. However, although the first prerequisition of finding good key frames was fulfilled by eliminating threshold dependence of the extraction procedure, there is the same concern on the two described methods as that on the method proposed in [4]: there is no comprehensive user study to prove the applicability of key frames extracted with these methods.

B. Video Highlights

Development of techniques for automated generation of preview sequences is a relatively new research area, and only a few works have been published recently. In [14], the *most characteristic* movie segments are extracted for the purpose of automatically producing a movie *trailer*. Movie segments to be included in such a trailer are selected by investigating low-level visual and audio features and by taking those segments which are characterized by *high motion* (action), *basic color composition* similar to average color composition of a whole movie, *dialog-like audio track*, and *high contrast*. Although the authors claim to obtain a good quality of movie abstracts, since "all important places of action are extracted," there is no user study to support that the segments selected using the above audio-visual features indeed capture the same material, which would be included into a manually produced trailer. As already mentioned in Section I, it is highly difficult to develop an automated system for extracting video highlights, due to the missing ground truth. Even if the extracted highlighting segments correspond to the video-content perception of one user, there may be another user for whom the obtain abstract is not or is only partially acceptable.

III. The Abstraction Approach Based on Cluster-Validity Analysis

The underlying principle of the video-abstraction method proposed in this paper is to remove the visual-content redundancy among video frames. The entire video material is first grouped into clusters, each containing frames of similar visual content. Taking again the dialog-like sequence from Section I (Fig. 2) as an example, the clustering process would group together all frames from all shots belonging to each of the content components A and B, resulting in this way in two clusters, one for each of the components. Then, by representing each cluster with its most representative frame, a set of key frames can be obtained that summarizes the given sequence. To obtain a summarizing preview sequence, it is sufficient to take the shots to which the extracted key frames belong as key segments and to concatenate them together.

Since the resulting number of key frames and key video segments in the preview sequence is dependent on the number of clusters, the problem of finding the most suitable abstract for a given sequence becomes the one of finding the optimal number of clusters, in which the frames of a video can be classified based on their visual content. The main difficulty here is that the optimal number of clusters needs to be determined automatically. To solve this, we apply known tools and methods of cluster validity analysis and tailor them to our specific needs.

As illustrated in Fig. 3, our abstraction method consists of three major phases. First, we apply N times a partitional clustering to all frames of a video sequence. The prespecified number of clusters starts at one and is increased by one each time the clustering is applied. In this way N different clustering possibilities for a video sequence are obtained. In the second step, the system automatically finds the optimal combination(s) of clusters by applying the cluster-validity analysis. Here, we also take into account the number of shots in a sequence. In the final step, after the optimal number of clusters is found, each of the clusters is represented by one characteristic frame, which then becomes a new key frame of a video sequence. The preview sequence is obtained by concatenating all video shots to which the extracted key frames belong. As will be explained at a later stage, we make the generation of a preview sequence dependent on the number of shots in a video.

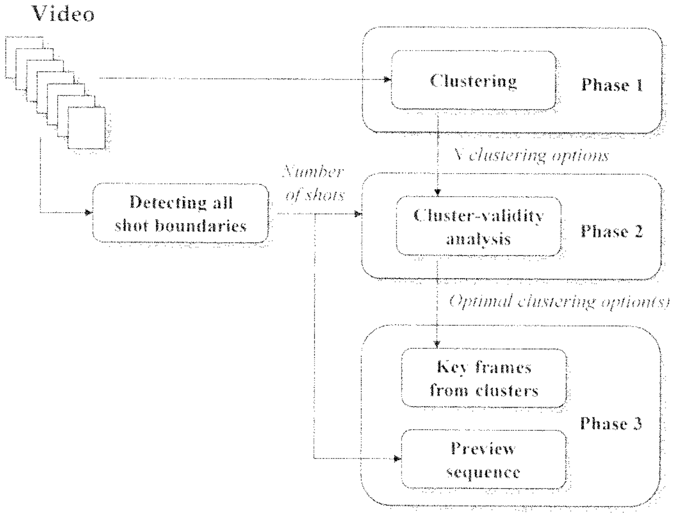


Fig. 3. Video-abstraction scheme based on cluster-validity analysis.

A. Clustering

The clustering process is performed on all video frames using the partitional clustering technique [11]. For this purpose, each frame k of a video sequence is represented by a D-dimensional feature vector $\vec{\phi}(k)$, consisting of features $\varphi_\nu(k)$. The feature vector can be composed using texture, color, shape information, or any combination of those. We wish to efficiently capture with key frames the changes in the visual material introduced, e.g., by camera motion, and to be relatively insensitive to object motion. Therefore, we have chosen a D-dimensional feature vector, consisting of the concatenated $D/3$-bin color histograms for each of the component of the YUV color space. Furthermore, since $\vec{\phi}(k)$ is easily computable, we also compensate in this way for an increased computational complexity of the overall abstraction approach due to the extensive cluster validity analysis but still achieve an acceptable frame content representation. The feature vector used is now given as

$$\begin{aligned}\vec{\phi}(k) &= \{\varphi_\nu(k)|\nu=1,\ldots,D\}\\ &= \left\{H_k^Y(1),\ldots,H_k^Y\left(\frac{D}{3}\right), H_k^U(1),\ldots,H_k^U\left(\frac{D}{3}\right),\right.\\ &\qquad \left. H_k^V(1),\ldots,H_k^V\left(\frac{D}{3}\right)\right\}.\end{aligned} \tag{4}$$

By taking into account the *curse of dimensionality* [12], we made the parameter D dependent on the sequence length and compute it as $S/5$ [12], where S is the number of frames to be clustered, and in this case also the number of frames in the sequence.

Since the actual cluster structure of the sequence is not known *a priori,* we first classify all frames of a sequence into 1-to-N clusters. Thereby, the number N is chosen as the maximum allowed number of clusters within a sequence by taking into account the sequence length. Although N can be understood as a thresholding parameter, its influence on the abstraction result is minimal. This is because we choose here N as much higher than the largest expectable number of clusters for a given sequence. The longer the sequence, the higher is the potential number of clusters for classifying its video material. We found the variation of N with the number of sequence frames S defined by the function (5) suitable for the wide range of sequences tested

$$N = N(S) = 10 + \left\lfloor \frac{S}{25} \right\rfloor. \tag{5}$$

When defining (5), we took into account that sufficient alternative options should be offered to the cluster validity analysis in order to obtain reliable results and that the number of options should increase with sequence length. On the other hand, the value N needs to be kept in limits, since the "noisy" clustering options become more probable with increasing number of clusters and can negatively influence the cluster validity analysis.

The clustering phase is followed by the cluster-validity analysis to determine which of the obtained N different clustering options, i.e., which number of clusters is the optimal one for the given sequence. In the following, we will explain this procedure in detail.

B. Cluster-Validity Analysis

For each clustering option characterized by n clusters $(1 \leq n \leq N)$, we find the centroids c_i $(1 \leq i \leq n)$ of the clusters by applying the standard k-means clustering algorithm on feature vectors (4) for all frames in the sequence. In order to find the optimal number of clusters for the given data set, we compute the *cluster separation measure* $\rho(n)$ for each clustering option according to [3] as follows:

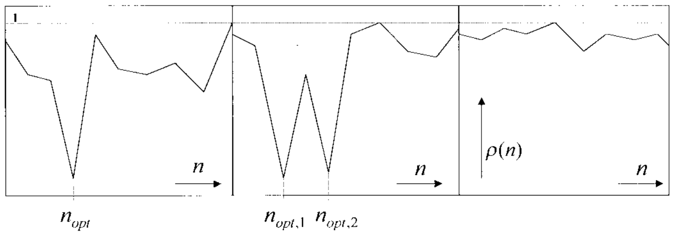
$$\rho(n) = \frac{1}{n}\sum_{i=1}^{n} \max_{1\leq j\leq n \wedge i\neq j}\left(\frac{\xi_i+\xi_j}{\mu_{ij}}\right), \qquad n \geq 2 \tag{6}$$

with the following parameters:

$$\begin{aligned}\xi_i &= \left\{\frac{1}{E_i}\sum_{k=1}^{E_i} |\vec{\phi}(k|k\in i) - \vec{\phi}(c_i)|^{\eta_1}\right\}^{1/\eta_1}\\ \mu_{\mathrm{ij}} &= \left\{\sum_{\nu=1}^{D} |\varphi_\nu(c_i) - \varphi_\nu(c_j)|^{\eta_2}\right\}^{1/\eta_2}.\end{aligned} \tag{7}$$

The better all of the n clusters are separated from each other, the lower is $\rho(n)$ and the more likely it is that the clustering option with n clusters is the optimal one for the given video material. E_i and ξ_i are the number of elements and the *dispersion* of the cluster i, respectively, while μ_{ij} is the *Minkowski metric* [5] of the centroids characterizing the clusters i and j. For different parameters η_1 and η_2, different metrics are obtained [3]. Consequently, the choice of these parameters has also certain influence on the cluster-validity investigation. We found the parameter setting $\eta_1 = 1$ and $\eta_2 = 2$ to give the best performance.

Note that the $\rho(n)$ values can only be computed for $2 \leq n \leq N$ due to the fact that the denominator in (6) must be nonzero. We now take all $\rho(n)$ values measured for one and the same sequence and for $2 \leq n \leq N$ and normalize them by their global maximum. Three different cases are possible for the normalized $\rho(n)$ curve, and they are illustrated in Fig. 4(a)–(c).



(a) (b) (c)

Fig. 4. Illustration of three possible cases for the normalized $\rho(n)$ curve.

Case 1: The normalized $\rho(n)$ curve is characterized by a pronounced global minimum at $n = n_{\text{opt}}$, as shown in Fig. 4(a). This can be interpreted as the existence of n_{opt} clear natural clusters in the video material with $n_{\text{opt}} > 1$. In this case, we assume a set of n_{opt} clusters to be the optimal cluster structure for the given video sequence.

Case 2: The normalized curve $\rho(n)$ has s distinct low values. This means that it is possible to classify the given video material into s different numbers of clusters with a similar quality of content representation. An example of this is illustrated in Fig. 4(b) for $s = 2$ with options containing $n_{\text{opt},1}$ or $n_{\text{opt},2}$ clusters.

Case 3: All values of the normalized $\rho(n)$ curve are high and remain in the same range (around 1), as illustrated in Fig. 4(c). This case can be interpreted in two ways: either there is no clear cluster structure within the given video material (e.g., an action clip with high motion) or the video sequence is stationary and can be treated as one single cluster. In the remainder of this paper, we will consider a sequence as *stationary* if there is no or only a nonsignificant camera or object motion (e.g., a zoom of a person talking, characterized by head and face motion). In general, if a $\rho(n)$ curve is obtained as shown in Fig. 4(c), the decision about the optimal cluster structure is made dependent on the detected number of shots in that sequence. This is explained more thoroughly in Sections III-B1 and III-B2.

Consequently, the problem of finding the optimal cluster structure for any video sequence given by the normalized $\rho(n)$ values for $2 \leq n \leq N$ is reduced to recognizing the most suitable of the three above cases. To do this, we first sort all the normalized values $\rho(n), 2 \leq n \leq N$, in ascending order, resulting in a sorted set $\rho_{sorted}(m), 1 \leq m \leq N-1$. Then, we introduce the reliability measure $r(m), 1 \leq m \leq N-2$, defined as

$$r(m) = \frac{\rho_{\text{sorted}}(m)}{\rho_{\text{sorted}}(m+1)}. \tag{8}$$

Last, we search for the value of the index m for which all values $r(m)$ are minimized. Two possible results of the minimization procedure are given by the following expressions:

$$\min_{1 \leq m \leq N-2} (r(m)) = r(1) \tag{9a}$$

$$\min_{1 \leq m \leq N-2} (r(m)) = r(s), \qquad s \neq 1. \tag{9b}$$

We will interpret these results for two different types of sequences, namely, sequences containing several video shots and those corresponding to single video shots.

1) Sequences Containing Several Video Shots: We first analyze the situation involving sequences that contain more than one video shot. If there is a pronounced global minimum of the $\rho(n)$ curve at $n = n_{\text{opt}}$, as shown in Fig. 4(a), the reliability vector $r(m)$ has its global minimum at $m = 1$. Therefore, the validity of (9a) is equivalent to the defined Case 1. Then, the optimal number of clusters is chosen as

$$n_{\text{opt}} = \min_{2 \leq n \leq N} (\rho(n)). \tag{10}$$

If (9b) is valid, the scope of possible options is constrained to either Case 2 or Case 3, whereby Case 3 can be considered less probable for the following two reasons: First, the probability of having a highly stationary content across several consecutive shots is low. Second, it is expected that there is sufficient distinction among the visual materials belonging to different shots of the sequence, such that several equally acceptable clustering options can be allowed. Therefore, we relate the validity of (9b) in case of complex sequences to the defined Case 2. That is, all cluster sets belonging to $\rho_{\text{sorted}}(i), 1 \leq i \leq s$, are taken as possible solutions for grouping the frames of a sequence.

2) Single Video Shots: For sequences consisting of only one video shot, the probability of finding a natural cluster structure containing more than one cluster is generally much lower than in complex sequences. This is because the changes of the visual content are continuous, mostly characterized by a camera/object motion without dominant stationary segments. For this reason, a large majority of $\rho(n)$ curves obtained for single video shots can be expected to correspond to the model in Fig. 4(c). This makes the reliable distinction between the stationary shots and the nonstationary ones having an unclear natural cluster structure crucial for obtaining a suitable abstract structure for single video shots.

If n_{opt} clusters are suggested by (10) for a given shot, and if that shot is stationary, the average intracluster dispersion $\overline{\xi}_{n_{\text{opt}}}$ computed over all n_{opt} clusters should be similar to the dispersion ξ_{one} computed when all frames of that shot are grouped into one cluster. Otherwise, the dispersion ξ_{one} can be assumed considerably larger than $\overline{\xi}_{n_{\text{opt}}}$. In view of this analysis, we define the decision rule (11) to distinguish stationary shots from the nonstationary ones. For this purpose, we first use (10) to find n_{opt} clusters for a given shot and compute the dispersion $\overline{\xi}_{n_{\text{opt}}}$. Then we also compute the dispersion ξ_{one} and compare both with $\overline{\xi}_{\text{ref}}$, which can be understood as the reference for the stationarity and is obtained by averaging dispersions measured for a large number of different stationary shots

$$|\xi_{\text{one}} - \overline{\xi}_{\text{ref}}| \underset{\text{stationary}}{\overset{\text{not stationary}}{\gtrless}} |\overline{\xi}_{n_{\text{opt}}} - \overline{\xi}_{\text{ref}}|. \tag{11}$$

If the shot is stationary, it is represented by only one cluster, including all frames of a shot. By nonstationary shots, we proceed with checking the evaluations (9a) and (9b). If (9a) is valid, n_{opt} is chosen as the optimal number of clusters, indicating that clearly distinguishable natural clusters exist

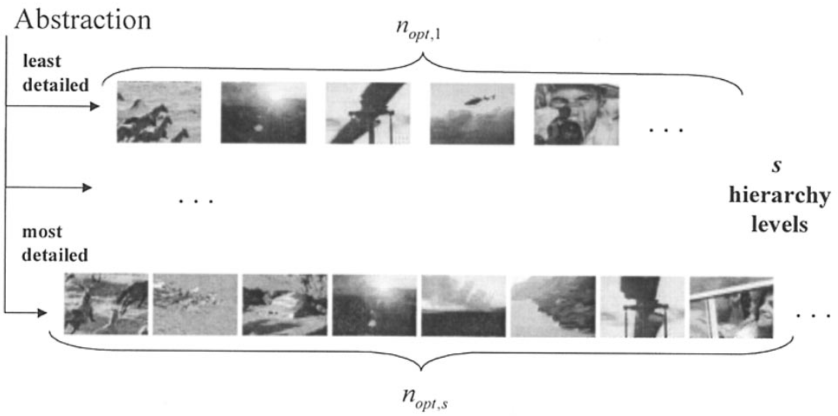


Fig. 5. Illustration of a hierarchical structure of a key-frame abstract by several clustering options.

within the shot. If (9b) is valid, we can assume that either there are several clustering options for the given shot or that no natural cluster structure can be recognized by the algorithm. The first possibility is relatively low due to a limited range of content variations within a shot of an average length. Therefore, the validity of (9b) for a single shot is related to an unclear cluster structure, which is difficult to represent. On one hand, one single cluster is too coarse, since variations of the visual content are present. On the other hand, choosing too many clusters would lead to an overrepresentation of the shot. For these cases, we found the smallest number of clusters proposed by (9b) as a good solution for this problem. Thus, from s clustering options suggested by (9b), we choose $n_{\min}$ clusters, defined by (12), to represent a single video shot with an unclear cluster structure

$$n_{\min} = \min_{1<i<s} (n_{\mathrm{opt,i}}). \tag{12}$$

C. Forming the Sequence Abstract

The process of finding the most suitable clustering option(s) for a given video sequence is followed by forming a video abstract. We will first discuss the procedure of building a set of key frames. One representative frame is chosen from each of the clusters and taken as a key frame of the sequence. As usual in the clustering theory, we choose for this purpose the cluster elements closest to cluster centroids. We find the key frame F_i of the cluster i by minimizing the Euclidean distance between feature vectors (4) of all cluster elements k and the cluster centroid c_i, that is

$$F_i \Leftarrow \min_{1<k<E_i} \sqrt{\sum_{\nu=1}^{D} |\varphi_\nu(k) - \varphi_\nu(c_i)|^2}. \tag{13}$$

If s different clustering options are found suitable for a sequence, key-frame sets extracted for each of the options are used to form a hierarchical key-frame structure, as illustrated in Fig. 5. Such a representation can provide a better interaction with video sequences having a complex structure.

While key frames are extracted in the same way for any arbitrary sequence, we make the forming of a preview sequence dependent on the number of shots contained in a video. This is because a preview is useful only in case of longer video, which contains more than X video shots (X specified *a priori*, application dependent). In our abstraction method, the video preview can be understood as a temporal extension of the key frames at the highest representation level (the clustering option with the smallest number of clusters in Fig. 5). Each shot, to which at least one extracted key frame belongs, is taken as a key video segment. These key segments are then concatenated to form the preview. The usage of entire shots for making a preview is preferred due to their complete contexts, e.g., a shot break mostly does not take place in the middle of a sentence. Such completeness makes the preview better understandable. An alternative to this is to use only shot fragments around key frames. However, the probability to have a complete context is considerably lower in this case. As an example, one could think of the abstraction of a longer dialog sequence using one image of each participating character for building the key-frame set and the corresponding video shots for building a small "dialog trailer." In such a preview, a couple of (most probably) complete sentences are desirable, spoken by each of the characters and revealing the dialog topic.

D. Application Scope

This last example of abstracting a reasonably structured video content illustrates the actual application scope of the abstraction method presented in this paper. Although this method can theoretically be applied to a video sequence of an arbitrary length, the sequences of interest in this paper are rather constrained to specific events, having a well-defined and reasonably structured content. The reason for this constraint is that long video sequences (e.g., full-length movies) are mostly characterized by an enormous visual content variety, which is difficult to classify in a number of distinct clusters and, consequently, difficult to represent by a limited number of key frames/segments.

Therefore, in order to be able to efficiently apply this approach to, e.g., a full-length movie, it is necessary to first segment the movie into well-structured, high-level fragments (scenes, story units, dialogs, etc.) [10]. After the segmentation is completed, our method can be applied to each of the movie fragments separately, as illustrated in Fig. 6. In this way, each fragment is represented by a structured collection of key

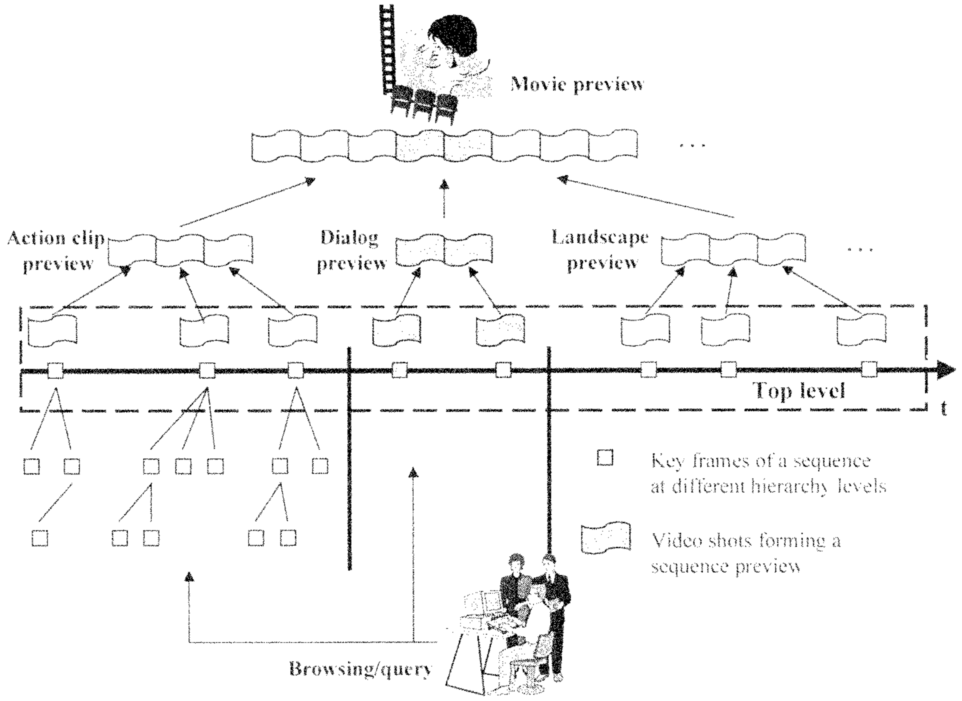


Fig. 6. A multilevel movie abstraction providing previews for each of movie segments and for the entire program, as well as episode-based comprehensive key-frame organization for browsing purposes or image queries.

frames and, eventually, by a suitable preview. Then, a preview of the entire movie can be formed too by concatenating the previews of its fragments. Using the scheme in Fig. 6, the user can easily follow the story, select the movie fragment of interest, browse through it or perform a pictorial query, look at its preview sequence, or simply at the preview sequence of the entire movie.

IV. Experimental Evaluation

In order to test the video-abstraction method presented in this paper, we concentrate here first on the evaluation of the proposed procedure for cluster-validity analysis, since both the key-frame sets and the preview sequences of a video abstract are directly dependent on the number and quality of obtained clusters.

We first tested the algorithm performance on sequences consisting of single video shots. For this purpose, we used 76 shots of a typical Hollywood-made movie and characterized them manually regarding the variations in their visual contents. The value of the parameter $\overline{\xi}_{\mathrm{ref}}$ from (11) was obtained experimentally as 0.0228, using a number of stationary shots containing different visual material and having different lengths, and can therefore be assumed generally valid. As illustrated in Table I, each of the shots belonging to the test set is assigned a description of how its content varies in time. From this description, the most suitable number of clusters for grouping all the frames of a shot is derived and used as a ground truth. For instance, a stationary shot should be assigned one cluster, or a shot with Q distinct stationary segments should be assigned Q clusters. For 66 shots (87%) of the test set, their frames were clustered in the same way as given by the ground truth.

TABLE I
A Fragment of the Test Set for Evaluating the Performance of the Cluster-Validity Analysis Algorithm for Single Shots

Shot 2:	Frames 42-286	stationary with minor object motion (1 cluster)
...		
Shot 10:	Frames 1582-1751	zoom (2 clusters)
...		
Shot 24:	Frames 4197-4358	two stationary camera positions (2 clusters)
...		
Shot 29:	Frames 5439-5776	three stationary camera positions (3 clusters)
...		
Shot 45:	Frames 7218-7330	camera panning (2 clusters)
...		
Shot 51:	Frames 8614-8784	stationary camera, followed by a strong zoom (2 clusters)
...		

In order to test the performance of the cluster-validity analysis algorithm for sequences containing several shots, we established a controlled test environment involving a set of sequences with clearly defined structure in terms of possibilities for clustering their frames. For each of these sequences, we estimated the suitable number of clusters for organizing their visual content and used this estimation as the ground truth. An indication about the algorithm performance can be found in Table II for the following test sequences used.

- *Sequence 1:* A dialog between two movie characters. Due to two fixed camera positions, two clearly defined clusters are expected, one for each of the characters.
- *Sequence 2:* Three movie characters in discussion with camera showing each of them separately and all together. Four clear clusters are expected.

TABLE II
OBTAINED CLUSTER STRUCTURES FOR LONG VIDEO SEQUENCES

TEST SEQUENCES	EXPECTED NUMBER OF CLUSTERS	EXPECTED CLUSTER STRUCTURE	OBTAINED NUMBER OF CLUSTERS	OBTAINED CLUSTER STRUCTURE
Sequence 1	2	Clear	2	Clear
Sequence 2	4	Clear	4	Clear
Sequence 3	2	Clear	2	Clear
Sequence 4	5	Clear	5,6	Unclear

- *Sequence 3:* Two major camera positions to be captured by two clear clusters.
- *Sequence 4:* Long sequence covering different visual material in a series. Five clear clusters are expected for sequence representation.

Although for the fourth sequence a clear cluster structure containing five clusters was expected, the algorithm suggested two possible clustering options. However, this was still acceptable, since the five clusters found corresponded to the expected ones and the option with six clusters contained the same clusters and an additional one, capturing a segment with object motion.

Based on the results of cluster-validity analysis, key-frame sets and preview sequences were formed. For each of the obtained clusters, a key frame was extracted using (13). Each time the obtained cluster combination corresponded to the one given by the ground truth, we also found the resulting key-frame set providing a good representation of the video content regarding the connection between objects and characters captured in key frames and the context of the story. This implies that frames nearest to cluster centroids are suitable to be used as key frames, and that the cluster-validity analysis is here the crucial step in making the video abstract. For the sequences listed in Table II, a preview was made by concatenating the shots, to which the extracted key frames belong. Regarding the qualities of these previews, the same conclusions can be drawn as in the case of key frames, since each key video segment is strongly related to its corresponding key frame(s).

V. CONCLUSIONS

Finding an alternative way of abstracting a video, such that the results are similar to those obtained manually, is a highly difficult task. This remains so even if we do not attempt to map human cognition onto the machine level, and if we constraint the applicability of the developed automated video-abstracting method only on the "objective" video summarization. In this paper, we presented an abstraction method having the objective of capturing the same "global" video material into the abstract and of keeping the similar abstract size, compared to the manual abstraction. In other words, the most important measure of good abstraction is not which key frames and key video segments are included into the abstract, as long as the same characteristic content components are captured. We found the principle of reducing the visual-content redundancy among video frames as a suitable practical way of reaching the posed objective. The largest problem to be solved was to find ways of automatically determining how many clusters are optimal for a given video. We solved this by developing an unsupervised procedure for cluster-validity analysis, presented in Section III.

REFERENCES

[1] F. Arman, A. Hsu, and M.-Y. Chiu, "Image processing on compressed data for large video databases," in *Proc. ACM Multimedia '93*, Anaheim, CA, 1993, pp. 267–272.

[2] Y. S. Avrithis, N. D. Doulamis, A. D. Doulamis, and S. D. Kollias, "Efficient content representation in MPEG video databases," in *Proc. IEEE Workshop Content-Based Access of Image and Video Database (in conjunction with CVPR '98)*, Santa Barbara, CA, 1998, pp. 91–95.

[3] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 224–227, Apr. 1979.

[4] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," draft version; see also *Proc. 6th ACM Int. Multimedia Conf.*, Bristol, U.K., 1998.

[5] H. P. Friedman and J. Rubin, "On some invariant criteria for grouping data," *J. Amer. Statist. Assoc.*, vol. 62, pp. 1159–1178, 1967.

[6] B. Furth, S. W. Smoliar, and H. Zhang, *Video and Image Processing in Multimedia Systems.* Norwell, MA: Kluwer Academic, 1995.

[7] P. Gresle and T. S. Huang, "Video sequence segmentation and key frames selection using temporal averaging and relative activity measure," draft version; see also *Proc. VISUAL '97*, San Diego, CA, 1997.

[8] B. Gunsel and A. M. Tekalp, "Content-based video abstraction," in *Proc. ICIP '98*, Chicago, IL, 1998, vol. III, pp. 128–132.

[9] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "A new method for key-frame based video content representation," *Image Databases and Multimedia Search*. Singapore: World Scientific, 1997, pp. 97–107.

[10] ——, "Automated high-level movie segmentation for advanced video retrieval systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 580–588, June 1999.

[11] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[12] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, vol. 2, P. Krishnaiah and L. N. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1982, pp. 835–855.

[13] A. Pentland, R. Picard, G. Davenport, and K. Haase, "Video and image semantics: Advanced tools for telecommunications," *IEEE Multimedia Mag.*, Summer 1994, pp. 73–75.

[14] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *J. Visual Commun. Image Representation*, vol. 7, no. 4, pp. 345–353, Dec. 1996.

[15] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, "A survey of thresholding techniques," in *Proc. CVGIP*, 1988, vol. 41, pp. 233–260.

[16] B. Shahraray and D. C. Gibbon, "Automatic generation of pictorial transcripts of video programs," in *Proc. IS&T/SPIE Digital Video Compression: Algorithms and Technologies*, San Jose, CA, 1995, pp. 512–519.

[17] X. Sun, M. S. Kankanhalli, Y. Zhu, and J. Wu, "Content-based representative frame extraction for digital video," draft version; see also *Proc. IEEE Multimedia Computing and Systems*, Austin, TX, 1998.

[18] SMASH Project. [Online]. Available HTTP: http://www-it.et.tudelft.nl/pda/smash.

[19] W. Wolf, "Key frame selection by motion analysis," in *Proc. IEEE ICASSP'96*, Atlanta, GA, 1996, pp. 1228–1231.

[20] W. Xiong, R. Ma, and J. C.-M. Lee, "A novel technique for automatic key-frame computing," in *Proc. IS&T/SPIE*, San Jose, CA, 1997, vol. 3022, pp. 166–174.

[21] M. M. Yeung and B. Liu, "Efficient matching and clustering of video shots," in *Proc. IEEE ICIP '95*, vol. I, pp. 338–341.

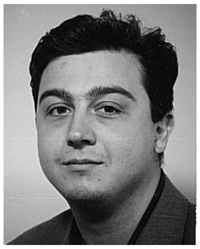
[22] H. Zhang, C. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," in *Multimedia Tools and Applications*. Norwell, MA: Kluwer, 1995, vol. 1, pp. 89–111.

[23] H. J. Zhang, C. Y. Low, S. W. Smoliar, and D. Zhong, "Video parsing, retrieval and browsing: An integrated and content-based solution," in *Proc. ACM Multimedia'95*, San Francisco, CA, Nov. 5–9, 1995, pp. 15–24.

[24] H. Zhang, J. Y. A. Wang, and Y. Altunbasak, "Content-based retrieval and compression: A unified solution," in *Proc. ICIP '97*, Santa Barbara, CA, 1997, vol. 1, pp. 13–16.

[25] H. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognit.*, vol. 30, no. 4, pp. 643–658, 1997.

[26] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proc. ICIP'98*, Chicago, IL, 1998, vol. 1, pp. 866–870.



Alan Hanjalic received the Diplom-Ingenieur (Dipl.-Ing.) degree in electrical engineering from the Friedrich-Alexander University, Erlangen-Nuremberg, Germany, in 1995. He currently is pursing the Ph.D. degree at Delft University of Technology, the Netherlands.

His research is in the area of visual-content analysis for advanced video-retrieval systems. From 1995 to 1998, he was a Researcher and Software Developer within the European ACTS Storage for Multimedia Applications Systems in the Home (SMASH) project. From May to September 1998, he was with Hewlett-Packard Laboratories, Palo Alto, CA, where his activities were concentrated on developing efficient video segmentation and abstraction techniques. In 1999, he was appointed an Assistant Professor in the Information and Communication Theory Group of the Delft University of Technology (Faculty of Information Technology and Systems). The scope of his activities includes teaching information theory and statistical signal processing at the undergraduate and graduate level as well as organizing and performing research in the area of video and image content analysis for browsing and retrieval applications in large multimedia data bases.

HongJiang Zhang (S'90–M'91–SM'97) received the B.S. degree from Zhengzhou University, China, and the Ph.D. degree from the Technical University of Denmark, both in electrical engineering.

He is a Research Manager at Microsoft Research, China. Prior to Microsoft, he was the Research Manager of Computational Video at HP Labs. During 1992–1995, he was with the Institute of Systems Science, National University of Singapore. He also worked at MIT Media Lab in 1994 as a Visiting Researcher. He has been actively engaged in research activities in the areas of video and image analysis, processing and retrieval, Internet multimedia, computer vision, and their application in consumer and enterprise applications. He is widely known in the multimedia research community for his pioneering work in video and image content analysis, representation, retrieval, and browsing. He is an author of two books, more than 70 referred papers and book chapters, 14 U.S. patents or pending applications, and numerous special issues of professional journals in content-based retrieval and Internet media. *Image and Video Processing in Multimedia Systems* (Norwell, MA: Kluwer, 1995), of which he was a co-author, was the first book that addressed content-based image and video retrieval research. He currently serves on the editorial boards of five professional journals and a dozen committees of international conferences. He is the Program Committee Co-chair of the ACM Multimedia Conference in 1999.