

RECENT ADVANCES IN VIDEO CONTENT ANALYSIS: FROM VISUAL FEATURES TO SEMANTIC VIDEO SEGMENTS

ALAN HANJALIC*, REGINALD L. LAGENDIJK[†] and JAN BIEMOND[‡]

*Department of Mediamatics, Delft University of Technology,
P. O. Box 5031, 2600 GA Delft, The Netherlands*

This paper addresses the problem of automatically partitioning a video into semantic segments using visual low-level features only. Semantic segments may be understood as building content blocks of a video with a clear sequential content structure. Examples are reports in a news program, episodes in a movie, scenes of a situation comedy or topic segments of a documentary. In some video genres like news programs or documentaries, the usage of different media (visual, audio, speech, text) may be beneficial or is even unavoidable for reliably detecting the boundaries between semantic segments. In many other genres, however, the pay-off in using different media for the purpose of high-level segmentation is not high. On the one hand, relating the audio, speech or text to the semantic temporal structure of video content is generally very difficult. This is especially so in “acting” video genres like movies and situation comedies. On the other hand, the information contained in the visual stream of these video genres often seems to provide the major clue about the position of semantic segments boundaries. Partitioning a video into semantic segments can be performed by measuring the coherence of the content along neighboring video shots of a sequence. The segment boundaries are then found at places (e.g., shot boundaries) where the values of content coherence are sufficiently low. On the basis of two state-of-the-art techniques for content coherence modeling, we illustrate in this paper the current possibilities for detecting the boundaries of semantic segments using visual low-level features only.

Keywords: Video Segmentation; Video Analysis; Video Retrieval; Video Browsing; Video Databases.

1. Introduction

In recent years, technology has reached a level where vast amounts of digital information are available at a low price. During the same time, the performance-versus-price ratio of digital storage media has steadily increased. Because it is easy and relatively inexpensive to obtain and store digital information while the possibilities to manipulate such information are almost unlimited, the *digital libraries* in the professional and consumer environment have grown rapidly. Examples are digital museum archives, Internet archives, image/video archives available to commercial

*E-mail: a.hanjalic@its.tudelft.nl

[†]E-mail: r.l.lagendijk@its.tudelft.nl

[‡]E-mail: j.biemond@its.tudelft.nl

service providers and private collections of digital information at home. All of these are characterized by a quickly increasing capacity and content variety.

With steadily increasing information volumes stored in digital libraries of various types, finding efficient ways to quickly retrieve information of interest becomes crucial. Since searching manually through gigabytes of unorganized stored data is tedious and time-consuming, the need grows for transferring information retrieval tasks to automated systems. Realizing this transfer in practice is not trivial and this especially for video and images. The main problem is that typical retrieval tasks such as “find me an image (or a video clip) showing a bird!” are formulated on a *cognitive level* according to the human capability of understanding the image or video content and analyzing it in terms of *semantic elements* like, for instance, the meaning (role) of objects, persons, sceneries, thematic (story) segments, meaning of speech fragments or the context of an image or a story in general.

Opposed to this, the information that can be extracted from an image or a video on the algorithmic or system level is much more “technical” than cognitive and consists of *low-level features*. Examples of these features are color distribution within an image, a video frame or a frame region, texture features (distribution of frequency coefficients in a textured region, wavelet coefficients, textural energy, contrast, coarseness, directionality, repetitiveness, complexity, auto-correlation, co-occurrence matrix, fractal dimension, auto-regressive models, stochastic models, edge distribution, shape/contour parameters and models, spatial relationships between lines, regions, objects, directional and topological relationships), motion vectors for frame regions providing the motion intensity and motion direction, audio features (pitch, frequency spectrum, temporal characteristics, etc.), speech characteristics (e.g., phonemes), etc. Also the temporal variations of the spatial and frequency features listed above can be investigated as “features of the features”.

To simulate the cognition-based image and video retrieval on the system level, i.e., to let the automated system extract elements of video semantics by analyzing low-level features of an image or a video, suitable algorithms operating on features need to be developed. These algorithms can be developed using the up-to-date techniques from image and audio analysis and processing, computer vision, statistical signal processing, artificial intelligence, pattern recognition and other related areas. The aimed parallelism between the cognition-based and feature-based image and video retrieval is illustrated in Fig. 1. With the objective to facilitate the interaction with large volumes of video material stored in emerging high-capacity digital video libraries, a vast diversity of feature-based algorithms for video content analysis was proposed in recent literature. These algorithms can be divided into four major categories,

- Video summarization,
- Extraction of semantically meaningful segments from a video,
- Semantics-based video classification,
- High-level segmentation.

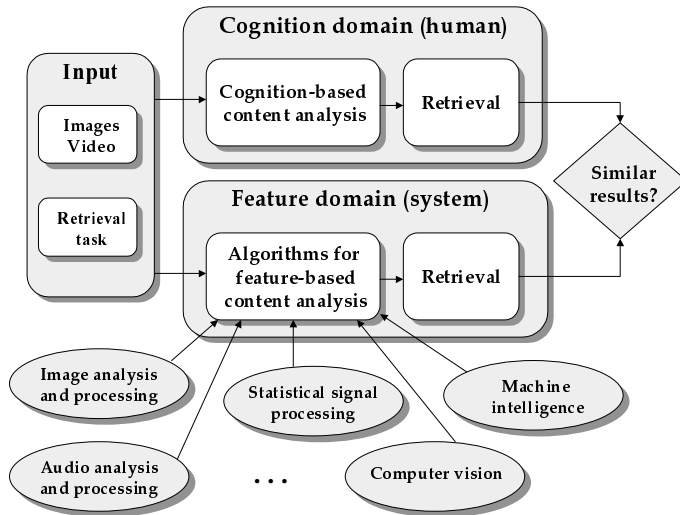


Fig. 1. Cognitive versus feature-based retrieval.

Since it aims at providing a first impression about a video to the user while keeping the information offered to the user as compact as possible, video summarization is an important step in increasing the efficiency of the interaction with a large-scale video database. Recently proposed summarization approaches address this problem from various aspects.^{1–11}

Extraction of semantically meaningful segments is basically a filtering of a video database and isolating the video segments that may be of interest for retrieval. For instance, using the approach by Saur *et al.*, specific action scenes like wide-angle and close-up views, fast breaks, steals or potential scores may be extracted from a basketball game.¹² In this way, the user does not have to watch the entire game (that may also contain some “boring” parts that are not worth watching) but can enjoy only the extracted highlights. To this category also belong algorithms for extracting the commercial breaks in various TV programs^{13–15} for detecting characteristic events in soccer broadcasts,¹⁶ for detecting events in a tennis game,¹⁷ for detecting dialogs, actions and story units in a movie¹⁸ and for extracting movie highlights.¹⁹

Semantics-based video classification is mostly performed in view of a number of pre-specified genres and aims at providing the top level of interaction between the user and a video database.^{20–25}

High-level segmentation is a content analysis step that is typical for video genres characterized by a clear sequential content structure. A sequence belonging to these genres can be modeled as concatenation of separate contexts — *semantic segments* — each of which is potentially interesting for retrieval. Examples of semantic segments are reports in a broadcast news program, episodes in movies, topic segments of documentary programs or scenes in a situation comedy. As illustrated in Fig. 2, a

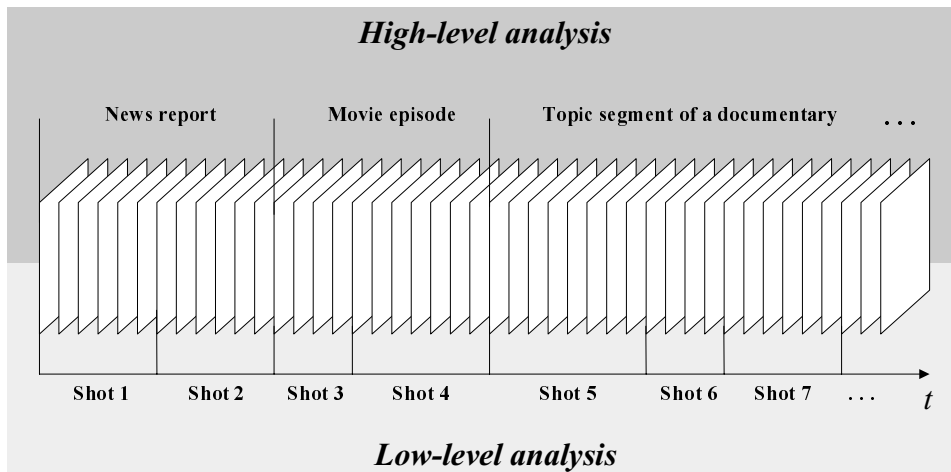


Fig. 2. Illustration of two different video analysis levels.

semantic segment can be understood as a concatenation of consecutive video shots that are related to each other with respect to their semantic content.

In view of the illustration in Fig. 2, we can conclude that the objective of high-level segmentation is to find *subsets* of all shot boundaries detected along a video such that the series of consecutive shots captured by shot boundaries belonging to these subsets correspond to the semantic segments of a sequence. Partitioning a video into semantic segments is typically approached by measuring the *coherence* of the content along neighboring video shots of a sequence. Content coherence can be computed based on the presence and temporal behavior of various low-level features in video shots considered. The segment boundaries are then found at places (e.g., shot boundaries) where the values of content coherence are sufficiently low.

In this paper, we focus on the problem of high-level video segmentation that is based on using visual low-level features only. In Sec. 2, we first discuss some major issues that are related to modeling of the content coherence along a video. Subsequently, we motivate in Sec. 3 our concentration on the usage of visual low-level features only and illustrate the current possibilities for content coherence modeling under this constraint on the basis of two state-of-the-art techniques. Section 4 concludes the paper.

2. Coherence of the Content in Neighboring Video Shots

Partitioning a video into shots can be considered an elementary or a low-level video-segmentation step. The reason for such a characterization is that this process as well as the obtained results do not depend on the actual content of the segmented video. Compared to the process of shot-boundary detection, high-level video segmentation is considerably more complex. Although both the low-level and high-level

segmentation have the same underlying principle — detection of coherence within the audiovisual material of a video clip and, consequently, of coherence breaks that determine the boundaries between these clips — the nature of this coherence and the level of difficulty for its detection are different in both cases.

On the one hand, the coherence within a single video shot is found to be mainly in terms of the continuity of visual features from one frame of a shot to the next following frame of that shot. A shot is taken by a single camera that zooms on an object, follows a moving object or pans along a scene. Due to a limited magnitude of motion and by a frame rate as high as 25 to 30 frames a second, each new frame contains a considerable portion of visual material from the previous frame. This type of coherence between consecutive frames may be explored by comparing neighboring frames with respect to their gray-level or color histograms,^{26–30} edge sets³¹ or by investigating the homogeneity of the motion vector field between the consecutive frames.³² The coherence ends at a shot boundary where camera starts to show some other scenes or objects with — in general — drastically different visual characteristics than in the previous shot. By suitably measuring the coherence between consecutive frames of a video and by applying suitable thresholds to isolate points of low coherence, shot boundaries may be detected rather successfully.^{33,34}

The coherence within semantic video segments is, on the other hand, primarily to be searched for as the continuity of the actual (semantic) video content from one video shot to the next following shot. In the following, we will refer to this type of coherence as *content coherence* as opposed to *visual-features coherence* in the case of shot-boundary detection. Then, semantic video segments can be defined as parts of a video where the content coherence values remain high. Consequently, the boundaries between neighboring segments can be expected at points at which the content coherence values decrease considerably. The content coherence among all video shots belonging to the same semantic segment may thus be considered as the full analogy to the coherence among the frames within a single video shot in terms of their visual characteristics. The question arises, however, how this content coherence may be detected, quantified and — similarly to the process of shot-boundary detection — separated from neighboring segments by segment boundaries. Can this be performed as easily as shot-boundary detection by simply measuring and thresholding suitable low-level features?

The difficulty of this problem may be seen on the example of the video clip taken from the Jurassic Park movie and represented by six characteristic frames in Fig. 3. The clip stretches over three semantic segments namely “discussion in a cottage” (Segment i), “meeting in a restaurant” (Segment $i + 1$) and “flying to the dinosaurs island” (Segment $i + 2$). Although the segment boundaries indicated in Fig. 3 are easy to determine manually, transferring this task to the feature level is not trivial. As can be seen from the “meeting” segment, the shots belonging to this segment are taken from different camera angles with different zooms and zoom targets. Further, the first shot showing the restaurant from the “street side” and the arrival of a person participating in the meeting is the introductory shot of the

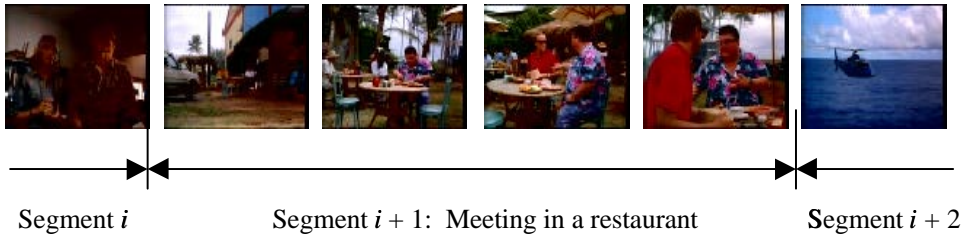


Fig. 3. Three consecutive semantic segments from the Jurassic Park movie.

segment although its similarity to other shots in terms of audiovisual features is very low. So, how can we obtain a high value of content coherence between the first segment shot and the rest of the shots of the “meeting” segment? Also, how can we guarantee low values of content coherence at the points where segment boundaries are indicated in Fig. 3?

As it will be seen from the approaches discussed in the forthcoming section obtaining the coherence values along a video that at least approximately depict semantic segments and their boundaries is indeed possible using low-level features. It also becomes obvious, however, that the feature-based high-level segmentation techniques must be based on certain assumptions regarding the content characteristics of a video sequence that belong to a certain genre (e.g., a movie, situation comedy, drama, news broadcast, documentary). These assumptions are necessary in order to be able to model the content coherence along a sequence as a function of low-level features. Consequently, the success of the segmentation process depends to a high degree on the validity of the assumptions made.

3. Coherence Modeling: Are Visual Low-Level Features Sufficient?

Combining text, audio, speech and visual information to detect the relation among neighboring shots is necessary for some video genres. A typical example of such a genre is a broadcast news program where report boundaries need to be determined. The report shown in Fig. 4 consists of five shots, each being represented by one frame. It can be easily seen that it is highly difficult (if not impossible) to recognize the thematic interrelation among the shots based on the analysis of their visual content only. For this reason, using information from the speech stream is unavoidable in the attempt to merge the shots in Fig. 4 into a report. As an illustration, in the news segmentation approach proposed by Hanjalic *et al.*,³⁵ first visual low-level features are used to detect the starting and ending points of all anchorperson shots. Several algorithms proposed in recent literature can be used for this purpose.^{36,37} Then, the analysis of the speech stream is performed and the middle points of all “silence” segments are marked. The time stamps in the middle of all “silence” segments and at the starting and ending points of all anchorperson shots denote the partitions of a pre-segmented news video. Each of the time stamps obtained is a



Fig. 4. The anchorperson shot introducing a report and the news shot series representing a report.

potential report boundary. Thus, the report-forming procedure is then basically the process of merging all elementary partitions that belong to the same topic.

Prior to the report-forming procedure, the speech stream of the news program is transferred into text. The report-forming procedure starts with filtering the text of elementary partitions and extracting only the words that are present in the topic database. This database covers a large number of topics and contains sets of keywords per topic. Each keyword is also assigned a weighting factor that quantifies the importance of that keyword for a certain topic. Weighting factors are useful since they allow, for instance, a long keyword list per topic and ease the usage of the same keywords for different topics.

Depending on which keywords are found in a certain elementary partition and what the weighting factors of these keywords are, topics from the database are assigned to each partition with the corresponding likelihood values. Then, the set of likelihood values per partition is thresholded in order to separate the most probable from least probable topics for each of the partitions. The threshold is computed by defining the “critical” number of keywords per topic in a partition and by requiring certain minimal values for weighting factors of the keywords found. In the next step, the behavior of the likelihood for each topic is investigated along all partitions of the news sequence and the likelihood of a topic per partition is adjusted depending on the likelihood for that topic in the surrounding partitions. In the final step, topics are assigned to each partition in view of the obtained likelihood values and the consecutive partitions being assigned the same topic are merged together into a report on that topic. Note that using the described procedure, multiple topics can be assigned to one partition which may be the case with the so-called “short-news” segments (quick overview of several topics without the appearance of an anchorperson in-between and with missing or unnoticeable “silence” speech segments).

Nevertheless, for many video genres other than broadcast news or a documentary, the pay-off in using different media for detecting the boundaries between

consecutive semantic segments is not high. On the one hand, relating the audio or speech features and even the keywords to the semantic temporal structure of video content is generally very difficult. This is especially so in “acting” video genres like movies and situation comedies. Some simple possibilities are, for instance, detection of “silence” fragments, distinguishing between speech and music, etc. On the other hand, the information contained in the visual stream of these video genres often seems to provide the major clue about the position of semantic segments boundaries.

While many proposals for segmenting broadcast news programs may be found in recent literature, e.g. in Refs. 35, and 38–41, not many approaches address the problem of partitioning other video genres into the corresponding semantic segments. In this section, we concentrate on two approaches that were developed for segmenting primarily movies and situation comedies.^{42,43} Both approaches use visual low-level features only to model the content coherence along a video.

3.1. Coherence modeling using shot recalls

Kender and Yeo model a video as a series of consecutive scenes and propose a technique for finding probable boundaries between consecutive scenes.⁴² For comparing visual similarity of shots k_m and k_n , the distance measure $D(k_m, k_n)$ has been introduced as

$$D(k_m, k_n) = \min_{i,j} D(f_{m,i}, f_{n,j}). \quad (1)$$

The distance $D(f_{m,i}, f_{n,j})$ measures the dissimilarity between frames $f_{m,i}$ and $f_{n,j}$ of a sequence and is used in the process of shot-boundary detection prior to the high-level segmentation process. Then, the measure $D(k_m, k_n)$ may be defined as the distance between the two most similar frames from shots k_m and k_n .

Using the shot dissimilarity measure from Eq. (1) and by taking into account the lengths and relative temporal positions of shots in a video sequence, Kender and Yeo model the content coherence as a continuous function that is evaluated at each shot boundary. The model is based on the assumption that the more the present shot and its nearby successors remind the viewer of the prior shots, the higher is the coherence at the time stamp of the present shot. This is indicated in Fig. 5(a). Opposed to this, the coherence value is low if the present shot and its nearby successors fail to remind the viewer of the previous content of a video sequence [Fig. 5(b)].

Based on the above assumptions, the coherence value is computed at each shot transition by checking the possibilities to establish *recalls* of shots preceding the boundary by the shots following the boundary. Actually, the coherence is measured at each shot boundary as the total recall of the shots older than the boundary by the shots newer than the boundary.

However, not all shots of a sequence are investigated for the possibility of establishing the recalls. This investigation is done only for the shots that are available

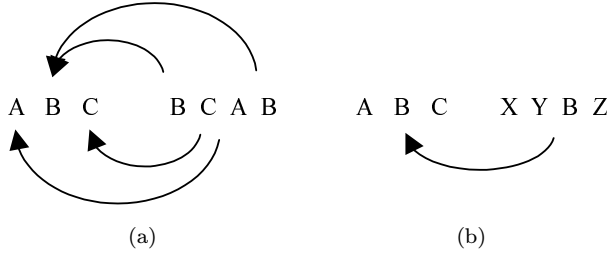


Fig. 5. (a) Good coherence (many future recalls), (b) bad coherence (few future recalls).

within a short-term memory buffer B . This buffer has adjustable length, moves along a sequence and surrounds in the form of a sliding window the shot boundary at which the total recall is computed. Furthermore, the buffering of the sequence material is modeled “as preserving the order of visual stimulus and as losing older frames uniformly throughout the buffer at the same aggregate rate as new frames are perceived”.⁴² Consequently, more recent frames are more likely to be recalled by the present frame while older frames are more likely to be lost due to buffer “leakage”. Also according to the buffer model, shots “shrink” as they get “older” which reduces their chances to be recalled.

To obtain the analytical expression for the coherence function, first, the recall $\text{SRecall}(k_m, k_n)$ between shots k_m and k_n is introduced as proportional to the function $\text{Sim}(k_m, k_n)$ describing their visual similarity and the function $\text{TR}(k_m, k_n)$ taking into account their lengths and their relative temporal positions within a video, that is:

$$\text{SRecall}(k_m, k_n) = \text{Sim}(k_m, k_n) \text{Tr}(k_m, k_n). \quad (2)$$

The similarity function $\text{Sim}(k_m, k_n)$ is obtained here simply by “inverting” the dissimilarity measure from Eq. (1), that is:

$$\text{Sim}(k_m, k_n) = 1 - D(k_m, k_n). \quad (3)$$

Then, the total recall of all the shots older than the boundary by all the shots newer than the boundary (note that only the shots within the buffer B are considered) is defined as:

$$\text{Recall}(k_i, k_{i+1}) = \sum_{m < i} \sum_{n < i+1} \text{SRecall}(k_m, k_n). \quad (4)$$

The coherence at the boundary between shots k_i and k_{i+1} is now computed as the total recall $\text{Recall}(k_i, k_{i+1})$ normalized by the maximum potential recall $\text{Ideal}(k_i, k_{i+1})$ possible at that boundary, that is:

$$\text{Coherence}(k_i, k_{i+1}) = \frac{\text{Recall}(k_i, k_{i+1})}{\text{Ideal}(k_i, k_{i+1})}. \quad (5)$$

The maximum potential recall $\text{Ideal}(k_i, k_{i+1})$ is computed similarly as $\text{Recall}(k_i, k_{i+1})$ except that $\text{Sim}(k_m, k_n)$ in Eq. (2) is fixed at its maximum value

of 1. The significant local minima of the coherence curve was measured along a sequence indicate the potential scene boundaries.

3.2. Coherence modeling using overlapping similarity links

Modeling the content coherence along a video sequence based on establishing links between related video shots is the path also followed by Hanjalic *et al.*⁴³ Their approach concentrates on movies and has the objective of partitioning an arbitrary movie sequence into episodes. Compared with the previously discussed method of Kender and Yeo, no recalls of the past shots are searched for but rather a sufficient percentage of the similar visual material in the shots coming ahead. Further, although the introduced principle of “overlapping similarity links” is indeed a model for coherence, Hanjalic *et al.* measure the content incoherence instead and use it for segmentation. Since, however, content incoherence is simply an inversion of coherence, we consider the objectives of both methods fully analogous.

Hanjalic *et al.* consider a hierarchical model of a movie structure which consists of three hierarchy levels namely *shots*, *events* and *episodes*. While shots are elementary “technical” temporal units of a video in general, an *event* is defined as the smallest semantic fragment of a movie. Such an event can be a dialog, an action scene or, generally, any series of shots unified by location or dramatic incident. However, an event does not need to be an unbroken series of consecutive shots; it can also alternate with another event. This is often used in the process of movie generation to represent several events taking place in parallel. Several alternating events are all together a good example of the highest semantic segment of a movie defined as an *episode*. There, all events are unified by the same *chronological time frame* of the story and form a rounded context which is in a certain sense separated from the neighboring contexts. An episode does not need to be related to several events; it can also concentrate on a single event. Since no shot within a movie is isolated but semantically it always belongs to a certain part of the story, each shot can be said to belong to one or to another episode. This implies that a movie can be understood as a *concatenation of episodes*.

The hierarchical model of the movie structure involving shots, events and episodes is illustrated in Fig. 6. There, the fragment i of the event j is denoted by E_i^j . The model shows how an episode is built up around one movie event or around several of them taking place in parallel. Thereby, a shot can either be a part of an event or it can serve for its “description” by, e.g., showing the scenery where the next or the current event takes place showing a “story telling” narrator in typical retrospective movies, etc. In view of such a distinction, shots of a movie are further referred to as either *event shots* or *descriptive shots*.

Capturing the content coherence among shots belonging to one and the same episode is directly related to the hierarchical model of movie structure from Fig. 6. Hanjalic *et al.* assume that an event is related to a specific location (scenery) and to certain movie characters. In other words, every now and then within an event,

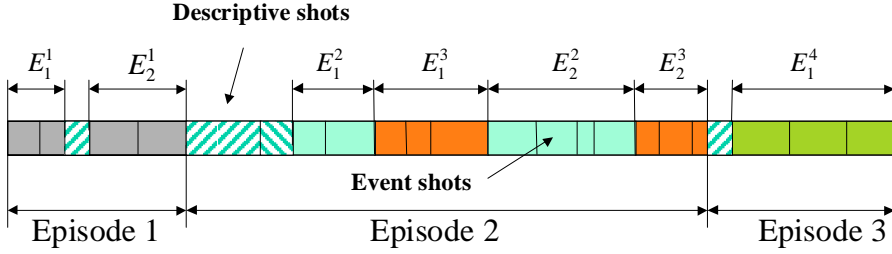


Fig. 6. Episodes 1 and 3 cover only one event and have a simple structure. Episode 2 covers two events, presented by their alternating fragments.

similar *visual content elements* (scenery, background, people, faces, dresses, specific patterns, etc.) appear and some of them even appear repeatedly. Since an episode is built around events, the same can be assumed for an episode as well; it is either related to only one event or to several of them alternating in time.

Assumption 3.1. An episode can generally be characterized by a global temporal consistency of its visual content, that is, by good matches of its visual-content elements found anywhere within a certain limited time interval.

According to this assumption, the content coherence among shots belonging to one and the same episode may be modeled by overlapping links that connect shots with similar visual content elements.

For comparing visual similarity of shots k and $k + l$, the distance measure $D(k, k + l)$ has been introduced as the result of a complex block-matching procedure. Prior to this procedure, shot-boundary detection has been performed and each shot has been represented by a number of key frames that optimally capture visual content of the shot. The block-matching procedure aims at discovering enough similar elements of the visual content in key frames of different shots. As content elements serve blocks that are taken from each key frame and represented each by its average color in the $L^*u^*v^*$ space. The dissimilarity $D(k, k + l)$ is then found as the average of the MSE distances of B percent of most similar block pairs.

Three different cases can be distinguished depending on the relation of the current shot k to the m th episode of a movie.

Case 3.1. Visual content elements from shot k_1 reappear (approximately) in shot $k_1 + p_1$. Then, shots k_1 and $k_1 + p_1$ form a linked pair. Since shots k_1 and $k_1 + p_1$ belong to the same episode(m), consequently, all intermediate shots also belong to m th episode,

$$[k_1, k_1 + p_1] \in \text{episode}(m) \quad \text{if} \quad p_1 \Leftarrow \min_{l=1, \dots, c} D(k_1, k_1 + l) < T(k_1). \quad (6)$$

Here, c is the number of subsequent shots (look-ahead distance) with which the current shot is compared to check the visual dissimilarity. The threshold function $T(k)$ specifies the maximum dissimilarity allowed within a single episode. Since the

visual content is usually time-variant, the function $T(k)$ also varies with the shot under consideration.

Case 3.2. There are no subsequent shots with sufficient similarity to shot k_2 , i.e., the inequality in Eq. (6) is not satisfied. However, one or more shots preceding shot k_2 link with shot(s) following shot k_2 . Then, the current shot is enclosed by a pair of shots that belong to m th episode, i.e.,

$$[k_2 - p_3, k_2 + p_2] \in \text{episode}(m)$$

$$\text{if } (p_3, p_2 > 0) \leftarrow \min_{i=1, \dots, r} \min_{l=-i+1, \dots, c} D(k_2 - i, k_2 + l) < T(k_2). \quad (7)$$

Here, r is the number of shots to be considered preceding the current shot k_2 (look-back distance).

Case 3.3. If for the current shot k_3 , neither Eq. (1) nor Eq. (2) is fulfilled and if shot k_3 links with one of the previous shots, then, shot k_3 is the last shot of episode(m).

To detect boundaries between episodes, one can in principle check Eqs. (6) and (7) for all shots in the video sequence. This however is computationally intensive and also unnecessary. According to Eq. (6), if the current shot k is linked to shot $k+p$ (link between shots (a) and (b) in Fig. 7), all intermediate shots automatically belong to the same episode, so, they need not to be checked. Only if no link can be found for shot k (shot (c) in Fig. 7), it is necessary to check whether at least one of r shots preceding the current shot k can be linked with a shot $k+p$ [for $p > 0$ as stated in Eq. (7)]. If such a link is found [link between shots (d) and (e) in Fig. 7], the procedure can continue at shot $k+p$; otherwise, shot k is at the boundary of episode(m) [shot (e) in Fig. 7]. The procedure then continues with shot $k+1$ for episode($m+1$).

In order to determine whether a link can be established between two shots, the threshold function $T(k)$ needs to be defined. This threshold is computed recursively

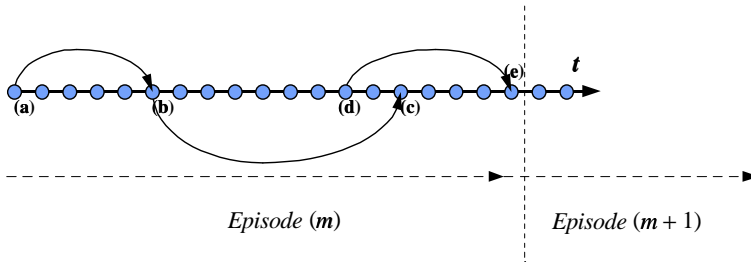


Fig. 7. Illustration of the episode boundary detection procedure. The shots indicated by (a) and (b) can be linked and are by definition part of episode(m). Shot (c) is implicitly declared part of episode(m) since shot (d) preceding (c) is linked to a future shot (e). Shot (e) is at the boundary of episode(m) since it cannot be linked to future shots nor can any of its r predecessors.

from already detected shots that belong to the current episode. For this purpose, the value of $\text{Incoherence}(k)$ at shot k is defined as the minimum of $D(k, n)$ found in Eq. (6) [or in Eq. (7) if Eq. (6) does not hold], that is:

$$\text{Incoherence}(k) = \begin{cases} D(k_1, k_1 + p_1) & \text{if Eq. (6) holds,} \\ D(k_2 - p_3, k_2 + p_2) & \text{if Eq. (7) holds.} \end{cases} \quad (8)$$

Then, the threshold function $T(k)$ is defined as:

$$T(k) = \frac{\alpha}{N_k + 1} \left(\sum_{i=1}^{N_k} \text{Incoherence}(k - i) + \text{Incoherence}_0 \right). \quad (9)$$

Here, α is a fixed parameter whose value is not critical between 1.3 and 2.0. The parameter N_k denotes the number of links in the current episode that have led to the current shot k while the summation in Eq. (9) comprises the shots defining these links. Essentially, the threshold $T(k)$ adapts itself to the content incoherence found so far in the episode. It also uses as a bias the last content incoherence value Incoherence_0 of the previous episode for which Eq. (6) or Eq. (7) is valid.

3.3. Discussion on performance

We address here the question of how efficient the detection of semantic segments may be if only visual features are considered in the partitioning process. Since in the specific video genres considered here — movies and situation comedies — the usage of other media is not likely to improve the segmentation process in a measure that would justify complex processes of audio, speech or text analysis, the discussion on the performance of the segmentation approaches described in this paper is basically the discussion on feasibility of segmenting the movies and situation comedies in general.

The performance of the segmentation approach proposed by Kender and Yeo is strongly dependent on the size of the short-term memory buffer B . With increasing buffer length B , more shots are remembered and recalled and the video tends to be perceived as being more coherent. Opposed to this, a shorter buffer contains less information which makes it more difficult to find recalls for incoming shots. The largest problem here is that there seems to be no “natural” buffer size at which the segmentation performance is optimized.

Kender and Yeo tested their segmentation approach on three sequences: the first hour of a science-fiction movie and two half-hour situation comedies. Due to the important role of the buffer length B , the tests concentrated on investigating the influence of B on the segmentation results. As a ground truth, scene boundaries were used that were obtained by human segmentation. First, B was set to the average shot length (there obtained as 124.2 frames). Subsequently, the buffer size was increased to values of 2, 4, 8, 16 and 32 times mean shot length. The results for all three test sequences have shown that the buffer size of 8 times mean shot

length provided the content coherence curve with local minima that most closely corresponded with the results of human segmentation.

Experimental results reported by Hanjalic *et al.* indicate that the modeling of the content coherence using overlapping links between shots with sufficient visual similarity is quite successful in detecting the episode boundaries. The performance of the proposed approach is however not perfect and this not because of the presence of small percentages of false alarms and missed episode boundaries. The largest imperfection is that the detected boundaries are in many cases only close to but not overlapping with the real episode boundaries. A typical example where this imperfection may occur is the first segment boundary indicated in Fig. 3. Since the first shot of the “meeting” segment is visually very different from other shots of that segment, the episode boundary may eventually be put after this shot if enough visual similarity is found between this shot and some preceding shot of the previous episode.

Hanjalic *et al.* evaluate their algorithm on two full-length movie sequences. Each shot was represented by two key frames taken from the beginning and end of a shot in order to capture most of its important visual content elements. To get a reliable idea about the positions of the actual episode boundaries, unbiased test subjects were asked to manually segment both movies and took into account only those boundaries registered by all test subjects. These boundaries are called *probable*. Then, the algorithm performed the segmentation of the movies for different values of parameters B (block percentage taken into account when comparing shots) and α (threshold parameter) and the automatically obtained boundaries were compared to the probable ones. Thereby, an automatically detected boundary, not registered by any of the test users, was considered as false.

Best performance for both movie sequences was obtained if 50% of blocks were considered in $D(k, k + l)$ and for the threshold multiplication factor α of 1.4. For these parameter values, 69% of probable boundaries were detected in average for both movie sequences with only 6% of false detections. It should be clear however that in order for a detected boundary to be proclaimed a “hit”, it is not necessary that it perfectly overlaps with a manually selected boundary. Due to the imperfection discussed above, certain distance from the real boundary is tolerated.

After investigating the missed 31% of probable boundaries, Hanjalic *et al.* found out that most of episodes which could not be distinguished from each other belong to the same global context (e.g., a series of episodes including a wedding ceremony, a reception and a wedding party). Therefore, the comprehensiveness of the boundary set obtained for $B = 50\%$ and $\alpha = 1.4$ was not strongly degraded by missed boundaries.

4. Conclusions

The need for tools capable of automatically managing large amounts of video data will steadily become larger with increasing volumes of video content stored in

emerging and growing video archives. A high level of sophistication is required for such tools since video material needs to be analyzed at the semantic level. Also, a large variety of algorithms for video content analysis needs to be developed due to an enormous diversity of video sequences with many different content characteristics.

Video sequences that can be modeled as concatenations of semantic segments take on a substantial share in the entire volume of video data that needs to be analyzed. Such sequences are, for instance, broadcast news programs, documentary programs with separated topic segments, movies divided into episodes and situation comedies divided into scenes. For sequences belonging to this class, efficient segmentation algorithms are required that are capable of detecting the boundaries between consecutive semantic segments. While, for instance, in the case of news broadcasts, the usage of different media is unavoidable for recovering the temporal semantic structure of the video content, the pay-off in considering other information than the visual one for segmenting movies and situation comedies is not high. Thus, segmentation algorithms there are dependent primarily on the usability of visual features for partitioning a video at the semantic level.

The current possibilities for segmenting movies and situation comedies using visual low-level features only were investigated in this paper on the basis of two state-of-the-art algorithms. Although both algorithms were based on a number of assumptions about the temporal content structure of a video and the connection of visual characteristics (features) of the video to this structure, they have both shown a high potential of using visual low-level features in recovering the semantic segments. Nevertheless, further development and refinement of these and similar methods is imperative in view of the crucial role that the visual content of a video has in determining the semantic temporal structure of a video.

References

1. Y. Gong and X. Liu, "Generating optimal video summaries," *Proc. IEEE Int. Conf. Multimedia and Expo 2000 (ICME 2000)* **3**, pp. 1559–1562.
2. P. Chiu, A. Girgensohn, W. Polak, E. Rieffel, and L. Wilcox, "A genetic algorithm for video segmentation and summarization," *Proc. IEEE Int. Conf. Multimedia and Expo 2000 (ICME 2000)* **3**, pp. 1329–1332.
3. C. Toklu, S.-P. Liou, and M. Das, "Videoabstract: A hybrid approach to generate semantically meaningful video summaries," *Proc. IEEE Int. Conf. Multimedia and Expo 2000 (ICME 2000)* **3**, pp. 1333–1336.
4. T. Syeda-Mahmood, S. Srinivasan, A. Amir, D. Ponceleon, B. Blanchard, and D. Petkovic, "CueVideo: A system for cross-modal search and browse of video databases," *Proc. IEEE Conf. Comput. Vision and Pattern Recognition 2000* **2**, pp. 786–787.
5. Y. Gong and X. Liu, "Video summarization using singular value decomposition," *Proc. of IEEE Conf. Comput. Vision and Pattern Recognition 2000* **2**, pp. 174–180.
6. A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "Efficient video summarization based on a fuzzy video content representation," *Proc. IEEE Int. Symp. Circuits and Systems 2000 (ISCAS 2000)* **4**, pp. 301–304.

7. N. Jeho and A. H. Tewfik, "Video abstract of video," *Proc. IEEE 3rd Workshop on Multimedia Signal Processing 1999*, pp. 117–122.
8. S. Uchihashi and J. Foote, "Summarizing video using a shot importance measure and a frame-packing algorithm," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1999* **6**, pp. 3041–3044.
9. N. Vasconcelos and A. Lippman, "A spatiotemporal motion model for video summarization," *Proc. IEEE Conf. Comput. Vision and Pattern Recognition 1998*, pp. 361–366.
10. A. Hanjalic and H. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Trans. Circuits and Systems for Video Technology, Special Issue on Object-based Video Coding and Description*.
11. L. Tiecheng and J. R. Kender, "A hidden Markov model approach to the structure of documentaries," *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries 2000*, pp. 111–115.
12. D. D. Saur, Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge, "Automated analysis and annotation of basketball video," *Proc. IS&T/SPIE* **3022**, February 1997.
13. R. Lienhart, C. Kuhmuench, and W. Effelsberg, "On the detection and recognition of television commercials," *Proc. IEEE ICMCS'97*, Ottawa, Canada, 1997.
14. T. McGee and N. Dimitrova, "Parsing TV programs for identification and removal of nonstory segments," *Proc. IS&T/SPIE Storage and Retrieval for Image and Video Databases VII* **3656**, January 1999.
15. C. Colombo, A. Del Bimbo, and P. Pala, "Retrieval of commercials by video semantics," *Proc. IEEE Conf. Comput. Vision and Pattern Recognition 1998* **2**, pp. 572–577.
16. Y. Gong *et al.*, "Automatic parsing of TV soccer programs," *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, May 1995.
17. G. Sudhir, J. C. M. Lee, and A. K. Jain, "Automatic classification of tennis video for high-level content-based retrieval," *Proc. Int. Workshop on Content-Based Access of Image and Video Database 1998*, pp. 81–90.
18. M. Yeung and B.-L. Yeo, "Video visualisation for compact presentation and fast browsing of pictorial content," *IEEE Trans. Circuits and Systems for Video Technology, Special Issue on Multimedia Technology, Systems and Applications*, October 1997.
19. S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, *J. Visual Commun. and Image Representation* **7**(4), 345 (1996).
20. S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," *Proc. ACM Multimedia'95*, San Francisco, 1995.
21. J. Huang, Z. Liu, and Y. Wang, "Joint video scene segmentation and classification based on hidden Markov model," *Proc. IEEE Int. Conf. Multimedia and Expo 2000 (ICME 2000)* **3**, pp. 1551–1554.
22. R. K. M. Rao, K. R. Ramakrishnan, N. Balakrishnan, and S. H. Srinivasan, "Neural net based scene change detection for video classification," *IEEE 3rd Workshop on Multimedia Signal Processing 1999*, pp. 247–252.
23. J. Huang, Z. Liu, Y. Wang, Y. Chen, and E. K. Wong, "Integration of multimodal features for video scene classification based on HMM," *IEEE 3rd Workshop on Multimedia Signal Processing 1999*, pp. 53–58.
24. A. Girgensohn and J. Foote, "Video classification using transform coefficients," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1999* **6**, pp. 3045–3048.
25. Y. Wang, J. Huang, Z. Liu, and T. Chen, "Multimedia content classification using motion and audio information," *Proc. IEEE Int. Symp. Circuits and Systems 1997 (ISCAS'97)* **2**, pp. 1488–1491.

26. A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," in *Visual Database Systems II*, eds. E. Knuth and L. M. Wegner, Vol. A-7 of IFIP Transactions A: Computer Science and Technology (North-Holland, Amsterdam, 1992), pp. 113–127.
27. B.-L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits and Systems for Video Technology* **5**(6) (1995).
28. B. Furht, S. W. Smoliar, and H. Zhang, *Video and Image Processing in Multimedia Systems* (Kluwer Academic Publishers, 1995).
29. H. Ueda, T. Miyatake, and S. Yoshizawa, "IMPACT: An interactive natural-motion picture dedicated multimedia authoring system," *Proc. CHI'91*, 1991.
30. K. Otsuji and Y. Tonomura, "Projection detecting filter for video cut detection," *Proc. ACM Multimedia'93*, 1993.
31. K. Mai, J. Miller, and R. Zabih, "A robust method for detecting cuts and dissolves in video sequences," *Proc. ACM Multimedia'95*, San Francisco, 1995.
32. A. Akutsu *et al.*, "Video indexing using motion vectors," *Proc. VCIP'92*, Boston, 1992.
33. N. Vasconcelos and A. Lippman, "A bayesian video modeling framework for shot segmentation and content characterization," *Proc. CVPR'98*, Santa Barbara, CA, 1998.
34. A. Hanjalic and H. Zhang, "Optimal shot boundary detection based on robust statistical models," *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, Florence, 1999.
35. A. Hanjalic, G. Kakes, R. L. Lagendijk, and J. Biemond, "DANCERS: Delft advanced news retrieval system," *SPIE/IS&T ELECTRONIC IMAGING 2001, Storage and Retrieval for Media Databases 2001*, San Jose, USA, 2001.
36. Y. Ariki and Y. Saito, "Extraction of TV news articles based on scene cut detection using DCT clustering," *Proc. ICIP'96* **3**, pp. 847–850, Lausanne CH, 1996.
37. A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Template-based detection of anchor-person shots in news programs," *Proc. IEEE Int. Conf. Image Processing (ICIP'98)*, Chicago, USA, 1998.
38. Q. Huang, Z. Liu, and A. Rosenberg, "Automated semantic structure reconstruction and representation generation for broadcast news," *Proc. IS&T/SPIE Storage and Retrieval for Image and Video Databases VII* **3656**, January 1999.
39. S. Eickeler and S. Muller, "Content-based video indexing of TV broadcast news using hidden Markov models," *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing 1999* **6**, pp. 2997–3000.
40. S. Boykin and A. Merlino, "Improving broadcast news segmentation processing," *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, Florence, 1999.
41. Y. Avrithis, N. Tsapatsoulis, and S. Kollias, "Broadcast news parsing using visual cues: a robust face detection approach," *Proc. IEEE Int. Conf. Multimedia and Expo (ICME 2000)* **3**, pp. 1469–1472.
42. J. R. Kender and B.-L. Yeo, "Video scene segmentation via continuous video coherence," *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Santa Barbara, June 1998.
43. A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video retrieval systems," *IEEE Trans. Circuits and Systems for Video Technology*, June 1999.



Alan Hanjalic (M'99) received the Diplom-Ingenieur (Dipl.-Ing.) degree in 1995 from the Friedrich–Alexander University in Erlangen, Germany, and the Ph.D. degree in 1999 from the Delft University of Technology, Delft, The Netherlands, both in Electrical Engineering.

From May to September 1998, he was with Hewlett–Packard Labs in Palo Alto, CA, as a Visiting Scientist and from December 2000 to January 2001 with British Telecom Labs, Ipswich, UK, as Research Fellow (awarded). Currently, he is an Assistant Professor at the Department of Mediamatics of the Delft University of Technology. The main research interest of Dr. Hanjalic is in the area of visual content management, that is, image and video content analysis for browsing and retrieval (query) applications. Dr. Hanjalic also investigates new image and video compression methodologies which can enable efficient performing of content-based operations on compressed images and video directly. Further, he is interested in subjective aspects of image and video analysis and retrieval that may be approached by using the techniques of affective computing. Related to these activities was the participation of Dr. Hanjalic in the European ACTS (AC018) project SMASH from 1995 to 1998. He authored and co-authored one book titled “Image and Video Databases: Restoration, Watermarking and Retrieval” (Elsevier 2000) and around 20 journal articles and conference papers.

Dr. Hanjalic serves in Program Committees of several conferences and as a Reviewer for various international scientific journals and conferences. He is an expert of the Belgian Science Foundation (IWT) and a member of IEEE and of IEEE Signal Processing Society.



Reginald L. Lagendijk (S'87-M'90-SM'97) received the M.Sc. and Ph.D. degrees in Electrical Engineering from the Technical University of Delft in 1985 and 1990 respectively. He became an Assistant Professor and Associate Professor in the Information and Communication Theory Group of the TU-Delft in 1987 and 1993 respectively. Since 1999, he has been Full Professor and Head of the Information and Communication Theory Group.

Prof. Lagendijk was a Visiting Scientist in the Electronic Image Processing Laboratories of Eastman Kodak Research in Rochester, New York in 1991. In 2000, he was Visiting Researcher at Microsoft Research, Beijing. He is author of the book “Iterative Identification and Restoration of Images” (Kluwer, 1991), and co-author of the books “Motion Analysis and Image Sequence Processing” (Kluwer, 1993) and “Image and Video Databases: Restoration, Watermarking and Retrieval” (Elsevier, 2000). He has been Associate Editor of the IEEE Transactions on Image Processing, and he is currently the Region Editor of Eurasip's

Signal Processing: Image Communications. Prof. Lagendijk is a member of the IEEE SP Society TC on Image and Multidimensional Signal Processing.

Research interests of Prof. Lagendijk include image and video compression, object-based compression, image quality measures, watermarking, image and video libraries, wireless multi-media communications, and image sequence restoration and enhancement. Prof. Lagendijk has been involved in several European projects on digital stereoscopic video communications, video indexing and retrieval, and film and video restoration. He is currently involved in the CERTIMARK project, a European project on certification of digital image and video watermarks. Prof. Lagendijk is Program Leader of the Delft University of Technology multidisciplinary research programme “Ubiquitous Communications (UbiCom)”.



Jan Biemond (M’80–SM’87–F’92) was born in De Kaag, The Netherlands. He received the M.S. and Ph.D. degrees in electrical engineering from Delft University of Technology, Delft, The Netherlands, in 1973 and 1982 respectively. Currently, he is a Professor in the Information and Communication Theory Group, Head of the Department of Mediamatics and Vice-Dean of the Faculty of Information Technology and Systems at Delft University of Technology.

His research interests include multidimensional signal processing, image enhancement and restoration, video compression (digital TV, stereoscopic TV, and HDTV) and video databases, and motion estimation with applications in image coding and computer vision. He has published extensively in these fields.

In 1983, he was a Visiting Professor at Rensselaer Polytechnic Institute, Troy, NY, and at Georgia Institute of Technology, Atlanta, GA. He was a Distinguished Lecturer of the IEEE Signal Processing Society for 1993–1994 and he is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE). He has served as the General Chairman of the Fifth ASSP/EURASIP Workshop on Multidimensional Signal Processing, held at Noordwijkerhout, The Netherlands, in September 1987. He was the General Chairman of the 1997 Visual Communication and Image Processing Conference (VCIP-97), San Jose, CA and he was Chairman of the 21st Symposium on Information Theory in the Benelux, May 25–26, 2000, Wassenaar, The Netherlands. He is the Scientific Editor of a series of books on “Advances in Image Communication” with Elsevier Science BV.

