Temporal Video Segmentation to Scenes Using High-Level Audiovisual Features

Panagiotis Sidiropoulos, Vasileios Mezaris, *Member, IEEE,* Ioannis Kompatsiaris, *Senior Member, IEEE,* Hugo Meinedo, Miguel Bugalho, and Isabel Trancoso, *Fellow, IEEE*

***Abstract*—In this paper, a novel approach to video temporal decomposition into semantic units, termed scenes, is presented. In contrast to previous temporal segmentation approaches that employ mostly low-level visual or audiovisual features, we introduce a technique that jointly exploits low-level and high-level features automatically extracted from the visual and the auditory channel. This technique is built upon the well-known method of the scene transition graph (STG), first by introducing a new STG approximation that features reduced computational cost, and then by extending the unimodal STG-based temporal segmentation technique to a method for multimodal scene segmentation. The latter exploits, among others, the results of a large number of TRECVID-type trained visual concept detectors and audio event detectors, and is based on a probabilistic merging process that combines multiple individual STGs while at the same time diminishing the need for selecting and fine-tuning several STG construction parameters. The proposed approach is evaluated on three test datasets, comprising TRECVID documentary films, movies, and news-related videos, respectively. The experimental results demonstrate the improved performance of the proposed approach in comparison to other unimodal and multimodal techniques of the relevant literature and highlight the contribution of high-level audiovisual features toward improved video segmentation to scenes.**

***Index Terms*—Audio events, scene transition graph, scenes, video segmentation, visual concepts.**

I. Introduction

VIDEO DECOMPOSITION into temporal units is an essential pre-processing task for a wide range of video manipulation applications, such as video indexing, nonlinear browsing, classification, and others. Video decomposition techniques aim to partition a video sequence into segments, such as shots and scenes, according to semantic or structural criteria. Shots are elementary structural segments that are defined as sequences of images taken without interruption by a single camera [1]. On the contrary, scenes are longer temporal segments that are usually defined as logical story units (LSUs): higher-level temporal segments, each covering either a single event (e.g., a dialog) or several related events taking place in parallel [2]. The close relation between video scenes and the real-life events depicted in the video make scene detection a key-enabling technology for advanced applications such as event-based video indexing; the latter has been gaining significant attention, as part of recent efforts toward experience and event-based multimedia manipulation [3]. Fig. 1(a) illustrates the relations between different temporal segments of a video.

Video segmentation to shots and scenes are two different problems that are characterized by considerably different degrees of difficulty. State-of-the-art shot segmentation techniques, detecting the presence of video editing effects such as cuts and fades with the use of low-level visual features, have been shown in large-scale experiments (e.g., TRECVID) to reach an accuracy that is close to perfect; this accuracy is deemed by the relevant community to be sufficient for any practical application [4]. On the contrary, scene segmentation is still an open research problem, with most approaches of the literature failing to take into account the semantics of the content in performing a task that by definition is based on semantic criteria; different consecutive parts of the video are assigned to the same scene, according to the literature, simply because they present similar low-level audiovisual properties, whereas it is much more than such low-level properties that make humans recognize (and request to consume, in applications such as retrieval) different scenes in a video.

In this paper, a novel approach to video temporal decomposition into scenes is presented. This builds upon the well-known technique of the scene transition graph (STG) [5], which it extends, and additionally exploits recent advances in sematic video analysis tasks in order to overcome the limitations of existing scene segmentation approaches. Initially, a new STG approximation that features reduced computational cost is introduced. This is important for ensuring the efficiency of a subsequent processing stage, which mandates the construction of multiple STGs. Then, a generalized STG-based (GSTG) technique is proposed for multimodal scene

Manuscript received October 11, 2010; revised January 31, 2011; accepted March 12, 2011. Date of publication April 7, 2011; date of current version August 3, 2011. This work was supported by the European Commission, under Contracts FP6-045547 VIDI-Video and FP7-248984 GLOCAL. This paper was recommended by Associate Editor Y. Rui.

P. Sidiropoulos is with the Informatics and Telematics Institute/Center for Research and Technology Hellas, Thermi-Thessaloniki 57001, Greece, and with the Center for Vision, Speech and Signal Processing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey GU2 5XH, U.K. (e-mail: psid@iti.gr).

V. Mezaris and I. Kompatsiaris are with the Informatics and Telematics Institute/Center for Research and Technology Hellas, Thermi-Thessaloniki 57001, Greece (e-mail: bmezaris@iti.gr; ikom@iti.gr).

H. Meinedo is with the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento, Lisbon 1000-029, Portugal (e-mail: hugo.meinedo@inesc-id.pt).

M. Bugalho and I. Trancoso are with the Instituto Superior Técnico, Lisbon 1049-001, Portugal and with the Instituto de Engenharia de Sistemas e Computadores, Investigação e Desenvolvimento, Lisbon 1000-029, Portugal (e-mail: miguel.bugalho@inesc-id.pt; isabel.trancoso@inesc-id.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2138830

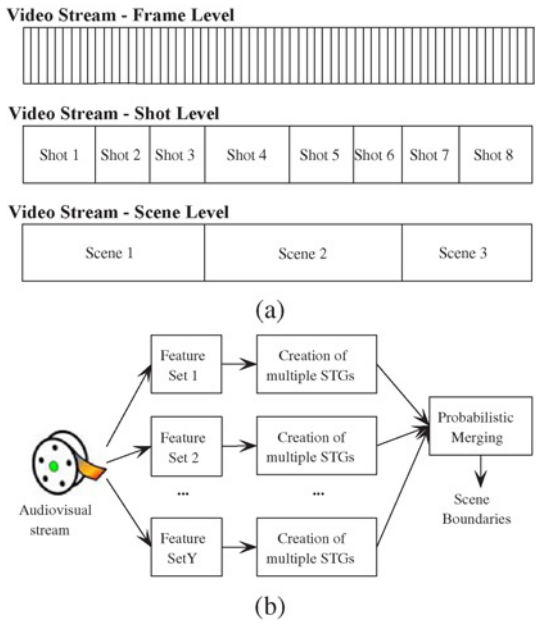

Fig. 1. (a) Video stream decomposition to frames, shots, and scenes. (b) Overview of the proposed approach for video segmentation to scenes.

segmentation. This is based on first constructing multiple STGs that separately exploit different audiovisual features for segmentation, using a new algorithm for the extension of STG to non-visual input, and second on using a probabilistic merging process to combine their results while at the same time diminishing the need for selecting and fine-tuning several STG construction parameters. In contrast to previous temporal segmentation approaches that employ mostly low-level visual or audiovisual features, the proposed technique jointly exploits low-level and high-level features automatically extracted from the visual and the auditory channel. The latter include model vectors that are made of visual concepts and audio events, previously unused in scene segmentation tasks. By taking into account several low-level and high-level features coming from multiple modalities and at the same time diminishing the need for heuristic parameter selection, the proposed approach becomes easily applicable to different video genres and delivers significantly more accurate results than previous methods, working only with low-level features. A broad overview of the proposed approach is given in Fig. 1(b).

The rest of this paper is organized as follows. The state-of-the-art in video segmentation to scenes is reviewed in Section II. The proposed fast STG approximation is introduced in Section III. The GSTG-based technique is developed in Section IV, followed in Section V by a presentation of the low-level and high-level audiovisual features used as part of GSTG in this paper. In Section VI, results from experiments and comparisons on three different datasets are reported, and our conclusions are drawn in Section VII.

II. Related Work

Many works on video segmentation to scenes have appeared in the last few years. A common characteristic of almost all of them is the assumption that each shot can belong to just one scene, thus scene boundaries are a subset of the video's shot boundaries. As a result, video temporal decomposition into scenes is typically based on some form of shot grouping; shots are identified using any one of the highly reliable shot segmentation approaches of the literature.

A. Scene Segmentation Techniques

The techniques of the relevant literature can be broadly classified into two classes, on the basis of the features that they use for representing the shots in the process of grouping them: unimodal techniques, which typically rely on visual information alone, and multimodal ones, typically combining visual and audio cues.

Unimodal techniques represent each shot with the use of low-level visual features. Global color features [e.g., hue-saturation-value (HSV) histograms] of selected keyframes are the most frequently used ones, although the use of color features of spatial regions of keyframes has also been proposed [6]. Color features are sometimes used in combination with motion [7] or structural information (e.g., shot length in [8]). The extraction of color and texture features directly from the compressed video stream, without selecting specific keyframes, has also been proposed [9]. Based on such shot representations, several algorithms have been used for grouping the shots into scenes. In [2], the keyframes of each shot are merged in one large variable-size image, called the shot image, and the similarity between blocks of different shot images is evaluated for the purpose of establishing links between shots. In [6], the similarity of shots is evaluated with the use of features extracted from selected regions of the keyframes, and editing rules from the film industry are also considered. Graph-based approaches have also received significant attention. In [5], pairwise color histogram similarities between keyframes are used for building a STG. In [7], a weighted undirected graph, with the weights expressing visual similarity and temporal proximity, is constructed and iteratively segmented into sub-graphs using normalized cuts [10]; another approach using normalized cuts is proposed in [11]. In [9], a similar temporal graph is constructed and an algorithm based on detecting shortest paths in it is used for determining the scene boundaries. In [8], a statistical approach, based on selecting an initial set of arbitrary scene boundaries and updating them using a Markov chain Monte Carlo technique, is presented. Finally, in [12], shot grouping is conducted by spectral clustering, without taking into account temporal proximity; the clustering outcome is used for assigning labels to the shots, and a sequence alignment algorithm is applied on the generated label sequences for identifying the scene boundaries.

Although such unimodal techniques are usually sufficient for clustering together shots characterized by pronounced visual similarities [e.g., Fig. 2(a)], the same does not stand true when the semantic relation between shots is indicated only by other means, e.g., by audio [Fig. 2(b)]. To address this shortcoming, the combined use of visual and audio cues has been proposed. Audio features typically used to this end include low-level ones such as short-time energy and zero-crossing rate, as well as intermediate-level results from the processing of the audio signal, such as audio segmentation, speech detection, and background conditions classification. In

(a) (b)

Fig. 2. Six keyframes of shots that belong to the same scene and (a) are characterized by pronounced visual similarities or (b) do not present significant visual similarities, but a relation between them is indicated by non-visual means (audio).

[13], an initial scene segmentation is performed using visual features alone, and adjacent scenes are further merged on the basis of low-level audio feature similarity. In [14], the video is decomposed to visual shots and audio segments; audio and visual segment boundaries are aligned to generate a set of candidate scene boundaries, which are accepted or discarded by further examining the audio changes. Similar in principle approaches, based on aligning the boundaries of visual and audio segments, are presented in [15] and [16]. In [17], scene changes are detected by evaluating the audio dissimilarity of adjacent shots only; a similar process is adopted in [18], where the notions of visual and audio attention are used for guiding the shot similarity evaluation. In [19], low-level color and audio features, together with face detection results, are used for computing a table of distances between the shots of a video that is exploited for clustering, while a weighted combination of audio and visual-similarity measures is used in [20]. In [21], a fuzzy k-means algorithm is introduced to segment the auditory channel into audio segments; scene breaks are identified when a visual shot boundary exists within an empirical time interval before or after an audio segment boundary. Learning-based methods are presented in [22]–[24]. Reference [22] proposed a statistical framework, which learns from a training set the probability of different shot features taking specific values on a scene boundary, and detects scene boundaries at local maxima of the likelihood ratio curve. In [23] and [24], audiovisual features are used as input to support vector machine (SVM) classifiers, which are trained to differentiate between two classes: scene-boundary and non-scene-boundary.

Common deficiency of the reviewed techniques is that they rely mostly on low-level audiovisual features. Although these are to some extent useful in evaluating the similarity of shots for the purpose of grouping them, there is a gap between the similarities that can be revealed by examining just low-level properties of the audiovisual signal and the semantic coherence that is desired of a scene. Another deficiency is that the combination of audio and visual information, which is evidently advantageous for scene segmentation, is typically performed either in a simplistic manner (e.g., simple temporal alignment of audiovisual segment boundaries) or with the use of learning-based techniques. The latter usually require large genre-specific manually segmented training sets.

B. Overview of the STG

In this section, a more detailed overview of the STG is given, since STG serves as the basis of the proposed approach. The STG is a technique introduced in [5]. It is an elegant unimodal technique, exploiting the visual similarity between keyframes of video shots to construct a connected graph; the cut-edges of this graph constitute the set of scene boundaries.

The STG construction starts with the generation of a segmentation S of the video B to non-overlapping visual shots as follows:

$$S = \{x_i\}_{i=1}^{N} \text{ where } x_i = \{f_k\}_{k=b_i}^{e_i},\ b_i < b_{i+1} \forall i$$

$$x_1 \cup x_2 \cup \ldots \cup x_N = B \qquad (1)$$

where f_k is the kth frame of the video, and b_i, e_i are the indices of the first and last frame of shot x_i, respectively. Two video shots are considered similar if they contain at least one pair of similar frames according to similarity measure $D(.,.)$ as follows:

$$D(x_i, x_j) = \min_{m,n} \left(D'(f_m, f_n) \right)$$

where

$$b_i \leq m \leq e_i \text{ and } b_j \leq n \leq e_j. \qquad (2)$$

In this equation $D'(f_m, f_n)$ is a measure of the similarity of frames f_m, f_n; typically, low-level features such as color histograms and distance measures such as L1 distance or histogram intersection are used. Although the similarity of all frames of both shots needs to be evaluated according to this criterion, a set of selected keyframes is often used instead, for reducing computational complexity.

The visual similarity values $D(x_i, x_j)$ between each pair of shots x_i, x_j in the video, providing that x_i, x_j are less than an empirical time threshold τ apart, are calculated and used for grouping shots that are similar (i.e., shots for which $D(.,.) < D_t$) into the same cluster. This clustering criterion requires each shot to be similar to every other shot in the same cluster. The order according to which the clustering proceeds is specified by $D(x_i, x_j)$; at any time, the most similar pair of shots is examined before all less similar ones. From the clusters and the temporal ordering of the shots, a STG is constructed, where nodes represent the shot clusters and a directed edge is drawn from a node to another if there is a shot represented by the first node that immediately precedes any shot represented by the second node. Finally, the "cut-edges" of the graph are identified. A cut-edge is defined as an edge which, if removed, results in two disconnected graphs. The collection of all cut edges constitutes the set of scene boundaries.

Among the advantages of the STG approach is that the evaluation of shot similarity is not limited to pairs of adjacent shots (thus, scenes characterized by repetitive patterns, such as dialogs, can be detected correctly), in contrast to several other unimodal or multimodal techniques. Among its disadvantages, though, is that it exploits only low-level visual features, it provides no support for combining heterogeneous feature sets, and similarly to most literature approaches it requires the heuristic setting of certain parameters (STG construction parameters D_t and τ).

III. Fast STG Approximation

The STG, as well as any other literature work reviewed above, performs shot grouping into scenes by examining whether a *link* exists between two shots; different criteria are used in each work for identifying potential pairs of shots (e.g., all shots lying within a temporal window) and for evaluating the presence or not of such links (e.g., the shots' HSV histogram similarity lying below a threshold). In this section, we use properties related to shot linking, such as shot linking transitivity and the fact that scenes are by definition convex sets of shots, to present an approximation to STG-based scene segmentation. This approximation limits the number of shot pairs whose possible linking needs to be evaluated and simplifies or renders obsolete other processing steps associated with the STG, thus allowing the faster detection of scene boundaries. The proposed approximation is not guaranteed to produce the exact same results as the original STG; nevertheless, the experiments in Section VI show that the performance differences are very small.

A. Definitions

Following the definition of the scene as a LSU [2], any scene segmentation process can be viewed as a clustering of shots into non-overlapping convex sets. Let us remind that in a totally ordered space, a set of points is convex if for every pair of points that belong to it, all points in between (according to the total order $<_o$ of the space) also belong to it. The shots of a video can be seen as defining a totally ordered 1-D space according to time, and scenes are indeed non-overlapping convex sets in this space: if two shots x_i, x_j belong to a single scene, then every shot x_m, $x_i <_o x_m <_o x_j$ also belongs to the same scene. The implication of this is that, having established a definitive link between shots x_i, x_j, it is redundant to look for links between any shots x_m, x_n if $x_i \leq_o x_m <_o x_n \leq_o x_j$, because of the convexity of the set that the link between shots x_i, x_j defines.

Considering the transitivity of shot linking, strictly speaking, shot linking is not a transitive relation. This can be seen with an example: assuming shots $x_i <_o x_m <_o x_j$, $D(.,.)$ being a shot similarity measure (e.g., HSV histogram difference) and $D(.,.) \leq a$ being the shot linking criterion, $D(x_i, x_m) \leq a$ and $D(x_m, x_j) \leq a$ do not necessarily mean that $D(x_i, x_j) \leq a$ also holds. However, viewing scene segmentation as the clustering of shots into non-overlapping convex sets, $D(x_i, x_m) \leq a$ and $D(x_m, x_j) \leq a$ means that x_i, x_m, x_j all belong to the same scene, and this is equivalent to establishing a shot link for the pair (x_i, x_j). For this, we will treat shot linking as a transitive relation in the sequel.

Based on the above considerations and assuming that a set $\mathcal{L}$ comprising K linked pairs of shots, $(x_{s_1}, x_{e_1}), \ldots, (x_{s_K}, x_{e_K})$, has been identified for a video B according to some linking criteria, we proceed with the following definitions.

Definition 1: A link between shots x_i and x_j is called a *trivial link* if there exists a $(x_{s_k}, x_{e_k}) \in \mathcal{L}$ such that $x_{s_k} \leq_o x_i$ and $x_{e_k} \geq_o x_j$.

Definition 2: Three shots x_i, x_m, x_j are said to define a *trivial double link* if both (x_i, x_m) and (x_m, x_j) belong to $\mathcal{L}$.

Algorithm 1 Primary set estimation

1. Initially, all pairs of shots (x_i, x_j), $x_i <_o x_j$, and $i, j \in [1, N]$, are marked as *valid pairs*; any pair that is examined in subsequent steps, and is not identified as linked, is automatically marked as an *invalid pair*. d is set to $N-1$ and i is set to 1.
2. d', d'' are set to zero.
3. If (x_i, x_{i+d}) is a valid pair, the presence of a link between these two shots is examined. If it is an invalid pair or no link is found: if $i + d < N$, this step is repeated after setting $i = i+1$, otherwise is repeated after setting $d = d-1$ and $i = 1$. This continues until a shot link is found or d becomes zero.
4. If pair (x_i, x_{i+d}) has been identified as linked, then starting from $d' = d$ and descending by step of one all valid pairs $(x_{i+d}, x_{i+d+d'})$ are examined sequentially for shot links, until a shot link is found or d' becomes zero.
5. If pair $(x_{i+d}, x_{i+d+d'})$ has been identified as linked, then starting from $d'' = d$ and descending by step of one all valid pairs $(x_{i+d+d'}, x_{i+d+d'+d''})$ are examined sequentially for shot links, until a shot link is found or d'' becomes zero.
6. If pair $(x_{i+d+d'}, x_{i+d+d'+d''})$ has been identified as linked, d' is set equal to $d' + d''$ and step 5 is repeated (without checking again if the condition of step 5 is satisfied); the algorithm oscillates between steps 5 and 6 until no further link can be found by these two steps.
7. If $d \neq 0$, $(x_i, x_{i+d+d'+d''})$ is added to the shot pairs that belong to the primary set of links; all pairs of shots (x, y), $x_i \leq_o x, y \leq_o x_{i+d+d'+d''}$ are marked as invalid pairs; i is set equal to $i + d' + d'' + 1$ [see Fig. 3(c)] and the algorithm returns to step 2. If $d = 0$, the algorithm terminates.

Definition 3: The set $\mathcal{L}$ is named *primary* if both no trivial links and no trivial double links exist in it.

Examples of a trivial link and a trivial double link are shown in Fig. 3. By introducing an algorithm that directly produces a primary set of links, i.e., avoids examining the existence of links that, given those already identified, would be trivial, we can reduce the computational cost associated with the detection of scene boundaries.

B. Shot Linking by Primary Set Estimation

Given the input video B that contains shots $x_1, x_2, \ldots, x_N$, as defined in Section II-B, a primary set of shot links can be directly estimated according to Algorithm 1.

It is evident that following this algorithm, no pair of shots is examined for the presence of a shot link more than once; also, as soon as a shot link is found (step 3), shot pairs potentially defining related trivial links are immediately excluded from further consideration. Related double trivial links are then looked for (steps 4–6) and, if found, are eliminated, further increasing the number of shot pairs that are excluded from subsequent processing. The resulting primary set of links $\mathcal{L}$ essentially defines a STG, with the convex sets of shots defined by the links in $\mathcal{L}$ serving as the nodes of the graph. With

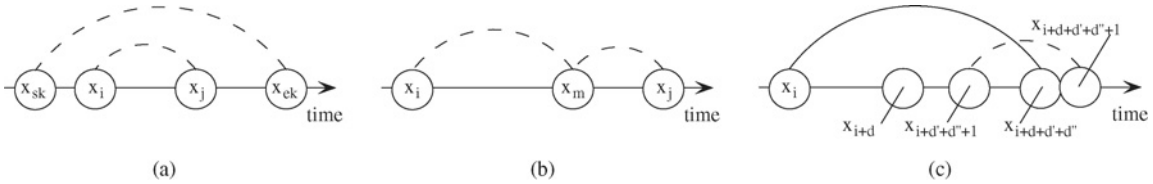

Fig. 3. Examples of trivial link, trivial double link, and illustration of a specific step of Algorithm 1. (a) If $(x_{s_k}, x_{e_k}) \in \mathcal{L}$, the link between shots x_i and x_j is a trivial link. (b) If $(x_i, x_m) \in \mathcal{L}$ and $(x_m, x_j) \in \mathcal{L}$, shots x_i, x_m, x_j define a trivial double link. (c) After Algorithm 1 finds and adds $(x_i, x_{i+d+d'+d''})$ to the primary set of links $\mathcal{L}$, as a result of finding in steps 2–6 a sequence of potential trivial double links $(x_i, x_{i+d}, x_{i+d+d'})$, $(x_{i+d}, x_{i+d+d'}, x_{i+d+d'+d''})$, and so on, index i in step 7 is set equal to $i + d' + d'' + 1$. In this way, Algorithm 1 then continues (going back to step 2) with examining whether pair $(x_{i+d'+d''+1}, x_{i+d+d'+d''+1})$ is linked.

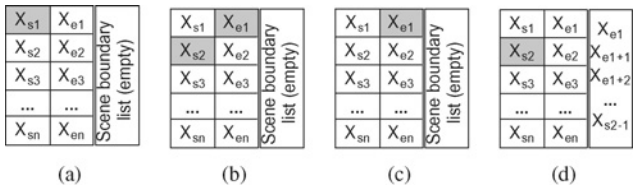

Fig. 4. Primary set of links $\mathcal{L}$ in tabular form, and example of browsing it so as to fill-in the list of scene boundaries. (a)–(d) Different stages of browsing the primary set of links.

respect to the maximum allowed temporal distance τ of linked shots, which is a parameter of the original STG, this can be integrated in Algorithm 1 simply by limiting accordingly the number of shot pairs that are marked as valid pairs in the first step of it.

Set $\mathcal{L}$ is parsed for detecting the scene boundaries as follows: all shot pairs that belong to it are ordered in tabular form, as shown in Fig. 4(a). Then, starting from the top-left cell:

1) if the current cell [Fig. 4(a)] belongs to the left column, we just move to one of the two neighboring cells [Fig. 4(b)] that corresponds to the shot that appears before the other one in B [according to the total order $<_o$, e.g., in Fig. 4(b), we will move to the x_{e1} cell if $x_{e1} <_o x_{s2}$, otherwise we will move to the x_{s2} cell];
2) if the current cell belongs to the right column [Fig. 4(c)], we move to the cell on the left column that is one row below the current one [Fig. 4(d)] and shots x_i of B that lie in between the two cells considered in this step [i.e., in the example of Fig. 4(c), shots for which $e1 \leq i < s2$] are added to the scene boundary list.

When the bottom-right cell is reached, the scene boundary list contains the last shot of each scene, i.e., the scene boundaries.

C. Computational Complexity Analysis

The main processing steps of the STG method for detecting scene boundaries and the corresponding steps of the proposed approximation of it are summarized in Table I. This table indicates that the proposed approximation is expected to deliver significant gains in computational complexity, since in it each main step of the STG is either simplified or becomes obsolete.

Specifically, with respect to the calculation of visual similarity values $D(.,.)$, the algorithm of the previous section refrains from checking a number of shot pairs for possible links after establishing a non-trivial link for shot pair (x_{s_k}, x_{e_k}). Assuming μ_k shots lie between shots x_{s_k} and x_{e_k}, this means that shot similarity measure $D(.,.)$ does not need to be computed for $\frac{(\mu_k+2)(\mu_k+1)}{2} - 1$ pairs of shots. For all K primary links in $\mathcal{L}$, the number of shot pairs for which $D(.,.)$ is not computed rises to $\sum_{k=1}^{K}(\frac{(\mu_k+2)(\mu_k+1)}{2} - 1)$, out of the $\frac{N(N-1)}{2}$ possible pairs of shots in B (assuming that $\tau \to \infty$). Consequently, the proportional computational complexity gain G from the use of the algorithm of Section III-B is $G = \frac{\sum_{k=1}^{K}(\mu_k)^2 + 3\sum_{k=1}^{K}\mu_k}{N(N-1)}$. This quantity is minimized when $\mu_k = \mu$, $\forall k \in [1, K]$, thus a lower bound for gain G is given by $G_{min} = \frac{\mu(\mu+3)K}{N(N-1)}$. Assuming, e.g., that out of $\frac{N(N-1)}{2}$ possible pairs of shots in B, non-trivial links are established for 5% of them (i.e., $K = 0.05\frac{N(N-1)}{2}$) and $\mu = 4$, the lower bound for gain G is 70%. This gain persists when additional limitations to the number of examined shot pairs are introduced (e.g., by $\tau \ll \infty$), providing that the non-trivial links continue to represent a reasonable portion of all the shot pairs that would otherwise be examined. Experiments indicate that 70% is indeed a typical value for G; this alone represents a speed-up by a factor of 3.

Considering the clustering of the shots, this step becomes obsolete in the proposed algorithm, whereas in the STG method this step involves, among others, the sorting of values $D(.,.)$ that have been calculated for each possible pair of shots. The latter process alone has average computational cost proportional to $\Lambda \log \Lambda$, where Λ denotes the number of shot pairs (when $\tau \to \infty$, $\Lambda = \frac{N(N-1)}{2}$). Finally, the parsing of the table of primary links, which is the last main step of the proposed algorithm, has very low computational cost (proportional to K, K being the number of primary links in $\mathcal{L}$). Although a direct theoretic comparison with the computational cost of algorithms for graph parsing is difficult, due to the different parameters affecting the latter (i.e., the number of nodes and edges of the graph, rather than K), the proposed parsing algorithm is intuitively expected to contribute to the overall speed-up of scene boundary detection.

IV. GSTG Method

The STG method for scene segmentation, regardless of whether the original algorithm of [5] or the fast approximation of Section III are used, is a method exploiting only low-level visual information for both the initial decomposition of the video stream to elementary video segments (shots) and for the

TABLE I
MAIN PROCESSING STEPS OF STG AND OF THE PROPOSED FAST APPROXIMATION OF IT

STG	Fast STG Approximation
Calculates visual similarity $D(.,.)$ for every pair of shots that do not exceed a specified temporal distance (τ).	Uses shot linking properties to further limit the number of shot pairs for which $D(.,.)$ needs to be calculated.
Clusters the shots (Section II-B); this requires sorting the shot pairs according to $D(.,.)$, and comparing the distances between all involved shot pairs for merging two clusters.	This processing step becomes obsolete; the primary links detected at the previous step directly define the shot clusters.
Parses the resulting graph (STG) to identify cut-edges.	Parses a much simpler structure (a table, as in Fig. 4).

similarity-based linking of them. In this section, we introduce: 1) a unimodal extension of STG to non-visual input, and 2) a method for combining unimodal STGs toward multimodal scene segmentation. Preliminary versions of these techniques have been introduced by the authors in [25] and [26].

A. Unimodal Extension of STG to Non-Visual Input

Non-visual features, e.g., low-level audio features, speaker diarization results, audio events, and others, can be used for providing two kinds of information in relation to the goal of video segmentation to scenes: 1) information about the similarity of two elementary video segments (e.g., shots), so as to allow for deciding whether the two segments are linked on not, and 2) binary information about the potential of a shot boundary to also be a scene boundary (i.e., allowed/non-allowed). The first possibility comes from using the non-visual features together with an appropriate similarity measure, analogously to the use of measure $D(.,.)$ for low-level visual features in the previous sections. The second possibility arises from the fact that the extraction of non-visual features from the audiovisual stream is typically accompanied by the definition of an appropriate decomposition of the stream to elementary segments. This decomposition in general does not coincide with the decomposition of the video to shots, and cannot be used by the STG in place of the latter, since this would lead to possible violation of the basic assumption that scene boundaries are a subset of the video's shot boundaries. It can however be used in combination with the decomposition to shots for limiting the number of shot boundaries that are treated as potential scene boundaries, with the help of simple semantic criteria.

For example, when performing speaker diarization for the purpose of describing each elementary video segment by a speaker identity (ID), a speaker segmentation of the audio stream is defined. The resulting speaker IDs can be mapped to the video shots, so that each shot is described by the histogram of speakers heard in it, and a suitable similarity distance can be defined for these shot feature vectors. The speaker segmentation of the audio stream can however provide additional binary information about the potential of a shot boundary to also be a scene boundary; the absence of a speaker change across a shot boundary, e.g., could be used as evidence that the two corresponding adjacent shots belong to the same scene.

In order to exploit such decomposition-based information when dealing with non-visual input, a few additional steps are introduced to the STG construction algorithm. Denoting S the decomposition of video into shots and S' the non-visual decomposition of the audiovisual stream to elementary segments, we proceed according to Algorithm 2.

Algorithm 2 Unimodal extension of STG to non-visual input

1. Adjacent segments of S' are merged according to similarity criteria set O, leading to segmentation S'_1.
2. The assumption that each segment of S'_1 can belong to just one scene is adopted. Based on this, adjacent shots of S are merged by eliminating shot boundaries that do not correspond to segment boundaries in S'_1, resulting in segmentation S_1. Evidently, if S'_1 and S coincide (e.g., when considering visual features), this processing step has no effect and S_1 also coincides with S.
3. Each segment of S_1 is described using appropriate features (e.g., in the case of speaker diarization results, by mapping speaker IDs to the segments of S_1, so that each segment is described by the histogram of speakers heard in it).
4. STG-based scene segmentation is performed (by means of either the algorithm of Section III or that of [5]), using segmentation S_1 instead of S as a starting point and replacing $D(.,.)$ with a similarity measure appropriate for the considered features.

In this extended algorithm, similarity criteria set O is used for correcting any over-segmentation errors in S', e.g., by merging two adjacent speaker segments of S' in case they are both assigned to the same speaker. Thus, the criteria in O are qualitative rather than quantitative and do not involve any distance measures or thresholds. For the second step, a temporal tolerance parameter is used when evaluating the correspondence of shot boundaries in S and segment boundaries in S'_1, to prevent minor misalignments from triggering the elimination of shot boundaries. Following this algorithm, various different STGs can be constructed for a single video, each based on different visual or non-visual features.

B. Combination of Unimodal STGs for Scene Segmentation

Despite the definition of the STG extension of Section IV-A, which in place of the typically employed low-level visual features can use different ones, the problem of combining multiple heterogeneous features remains. At the same time, it has been experimentally found that regardless of the considered features, the estimated scene boundaries depend significantly on the value of parameters that are inherent to the STG construction process, namely, the temporal distance τ and the similarity threshold D_t. In order to combine multiple

heterogeneous features for scene segmentation and simultaneously reduce the dependence of the proposed approach on parameters, we propose a probabilistic technique that involves the independent creation of multiple STGs of each type, where a "type" means here an STG that uses a specific set of features (e.g., just low-level visual ones). Specifically, following the creation of multiple ($\mathcal{P}$; $\mathcal{P} \gg 1$) STGs of type y, using a different set of randomly selected parameter values (τ, D_t) for each of them, the scene boundaries according to each STG (cut-edges) are extracted. Then, for every pair of adjacent shots x_i and x_{i+1}, the number p_i^y of STGs that have identified the boundary between these shots as a scene boundary divided by the total number of generated STGs of this type is calculated and used as a measure of our confidence on this shot boundary also being a scene boundary, based on the features that STG type y employs. The same procedure is followed for all different types of STGs, i.e., for all different features. Subsequently, these confidence values are linearly combined to result in a cumulative confidence value p_i as follows:

$$p_i = \sum_y w_y \cdot p_i^y \tag{3}$$

where w_y are global parameters that control the relative weight of each type of STGs, i.e., of each type of features ($\sum w_y = 1$). Finally, all shot boundaries (x_i, x_{i+1}) for which p_i exceeds a threshold as follows:

$$\Gamma = \{(x_i, x_{i+1}) | p_i > T\} \tag{4}$$

form the set Γ of scene boundaries estimated by the proposed approach. The advantage of this probabilistic approach is that multiple features are combined and at the same time the need for experimentally setting STG construction parameters τ, D_t is alleviated. Additionally, instead of introducing some feature combination weights in $D(.,.)$, which would turn these into difficult to optimize STG construction parameters, weights w_y that combine the results of already constructed STG are introduced; these weights are easy to optimize using least squares estimation (LSE). An illustration of the resulting GSTG, using the four different sets of features introduced in Section V, is given in Fig. 5.

V. Audiovisual Features for GSTG

In this paper, four different sets of features are combined and evaluated as part of the GSTG method. Some of them have been previously used for video segmentation to scenes, while others are novel ones, at least with respect to their use in such a task. Overall, the employed feature sets are: 1) typically used low-level visual features (HSV histograms); 2) model vectors constructed from the responses of a number of visual concept detectors; 3) typically used audio features (background conditions classification results, speaker histogram); and 4) model vectors constructed from the responses of a number of audio event detectors. For the above four feature sets, index y (3), which denotes the type of constructed STGs according to the features used for their construction, takes values V, VC, A, and AE, respectively.

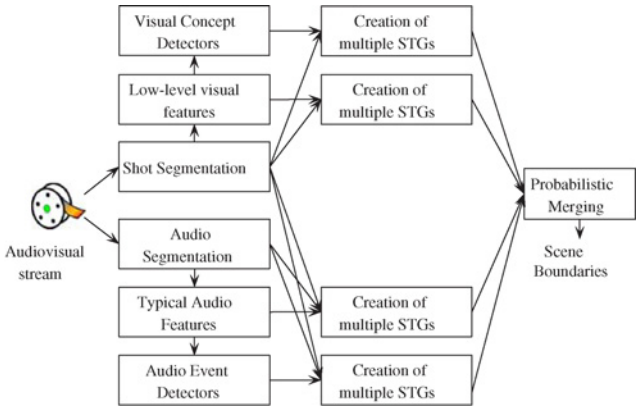

Fig. 5. Block diagram illustrating the GSTG, for the different types of features/STGs used in this paper. The audiovisual stream is decomposed into shots and audio segments, and different visual and audio features are extracted. The features and the initial segmentation results are used to generate four different types of STGs, whose results are subsequently merged according to a probabilistic merging process in order to estimate the final scene boundaries.

A. Typical Visual Features

The HSV histograms of a few keyframes of each shot, or very similar representations, have been extensively used in the relevant literature (e.g., [5]) and are also used in this paper, together with the L1 distance as a shot similarity measure $D(.,.)$.

B. Visual Concept-Based Model Vectors

Model vectors are constructed from the responses of trained visual concept detectors and are used in this paper as high-level visual features. Model vectors were originally proposed for the task of image and video retrieval [27], [28].

The visual concepts used in this paper are the 101 concepts defined on the TRECVID 2005 dataset (made of Broadcast News videos) as part of the Mediamill challenge [29]. These concepts range from relatively abstract ones (e.g., "outdoor") to very specific ones, such as names of individuals that were frequently in the news at that time (e.g., "B. Clinton"). Using them and the training portion of the annotated TRECVID 2005 dataset, a concept detector is trained for each concept separately. This detector combines a set of MPEG-7 features (color structure, color layout, edge histogram, homogeneous texture, and scalable color) [30] with a bag-of-words feature vector with the use of SVM classifiers. More details on the implementation of the concept detectors and the utilized visual concepts can be found in [31].

The application of J_v different trained visual concept detectors on a keyframe f of a shot results in J_v degree of confidence values, which can be expressed as a vector $\phi(f)$ as follows:

$$\phi(f) = [\phi_1(f), \phi_2(f), \ldots, \phi_{J_v}(f)]. \tag{5}$$

This vector essentially represents keyframe f in the semantic space defined by the J_v concepts. Subsequently, in order to take into account the results of concept detection in more than one keyframes per shot and also alleviate qualitative differences between different detectors, the shot representation

vector $\phi(x)$ is defined as follows:

$$\phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_{J_v}(x)], \;\; \phi_\gamma(x) = \frac{\max_{f \in x}\{\phi_\gamma(f)\}}{\max_{f \in B}\{\phi_\gamma(f)\}}. \tag{6}$$

The denominator in the second part of (6) denotes the maximum value of the γth concept detector across all keyframes of the examined video.

The definition of a shot similarity measure using the model vectors is based on the requirement that not only the difference of values $\phi_\gamma(x)$ between two shots but also the absolute values $\phi_\gamma(x_i)$ and $\phi_\gamma(x_j)$ themselves, should affect shot similarity. The rationale behind this is that, for the γth detector, two shots receiving similarly high confidence values is a strong indication of their semantic similarity (i.e., they are both likely to depict the γth concept). On the contrary, the same shots receiving similarly low confidence values is an indication neither in favor nor against their semantic similarity; it merely suggests that the γth concept (out of a large number J_v of concepts) is not depicted in either of the two shots. The commonly used L1 or other Minkowski distances do not satisfy the above requirement, since they depend only on the difference of the values. Instead of it, a variation of the Chi-test distance is employed in this paper, defined as follows:

$$D(\phi(x_i), \phi(x_j)) = \sqrt{\sum_{\gamma=1}^{J_v} \frac{(\phi_\gamma(x_i) - \phi_\gamma(x_j))^2}{\phi_\gamma(x_i) + \phi_\gamma(x_j)}}. \tag{7}$$

It should be noted that the TRECVID 2005 dataset, on which the visual concept detectors were trained, is a concept-annotated dataset extensively used for concept detector training and evaluation, and it is completely unrelated to the two test datasets used for experimentation in Section VI.

C. Typical Audio Features

Audio features typically employed for video segmentation to scenes include low-level features (e.g., short-time energy, zero-crossing rate) and somewhat higher-level ones (e.g., the results of audio segmentation, background conditions classification, speaker clustering, and others).

In this paper, we extract audio features by performing audio segmentation, classification according to background conditions, and speaker diarization [32], [33]. Background classification considers three classes: noise, silence, and music. Speaker diarization identifies speaker homogeneous segments in the audio stream and further assigns a speaker identity to each, after clustering them. The result of this process is the partitioning of the audiovisual stream into audio segments, each of which carries a background class label and, in case it also includes speech, a speaker ID as well.

For exploiting these features, criteria set O (Section IV-A) is defined as two adjacent audio segments sharing the same background conditions and speaker ID labels; the feature used for describing each segment of segmentation S_1 (an intermediate result of the algorithm of Section IV-A) is a speaker identity distribution, defined as follows:

$$H(x) = [H_1(x), H_2(x), \ldots, \; H_\Theta(x)] \tag{8}$$

where x denotes in this equation a temporal segment of segmentation S_1 rather than an original shot in S, and Θ is the total number of speakers in the video as per the speaker diarization results. $H_\theta(x)$ is defined as the fraction of time that speaker θ is active in video segment x over the total duration of the same segment. Similarly to the HSV histograms, the L1 distance is used as a segment similarity measure $D(.,.)$.

D. Audio Event-Based Model Vectors

Audio events are the audio equivalent to visual concepts. An audio event is defined as a semantically elementary piece of information that can be found in the audio stream, such as telephone ringing, dog barking, music, child voice, traffic noise, and others. Audio events are detected with the use of trained audio event detectors that rely on machine learning, outlined as follows.

1) Classification using SVMs as described in [34] for 61 audio events, e.g., dog-barking, siren, applause, explosion.
2) Classification using multilayer perceptrons or Gaussian mixture models as described in [35] for 14 audio events, e.g., male speaking, speech with noise background, and music.

The complete list of considered audio events is given in Table II.

Similarly to the way the results of visual concept detectors are used in this paper, the responses of the audio event detectors (confidence values for the presence of a specific audio event in a given audio segment) are used to build audio event-based model vectors as follows:

$$\psi(x) = [\psi_1(x), \psi_2(x), \ldots, \psi_{J_a}(x)] \tag{9}$$

where x denotes again a temporal segment of segmentation S_1, produced using the same criteria O as in the previous Section V-C. For the reasons discussed in Section V-B, the variation of the Chi-test distance introduced in (7) is also used here for comparing audio segments according to their audio event-based model vectors. Finally, it should be noted that, similarly to the visual concept detectors, the audio event detectors were trained on an annotated audio event corpus [34], [35] completely unrelated to the two test datasets used for experimentation.

VI. Experimental Results

A. Datasets and Evaluation Measures

For experimentation, two datasets were used in all experiments, while a third one was additionally used in a few experiments for showing the applicability of the proposed approach to a certain type of news videos. The first dataset is made of 15 documentary films (513 min in total) from the collection of the Netherlands Institute for Sound and Vision,[1] also used as part of the TRECVID dataset in the last few years. The second one is made of six movies (643 min in total). Application of the shot segmentation algorithms of [36] and [37] (for abrupt

[1] http://instituut.beeldengeluid.nl.

TABLE II
LIST OF AUDIO EVENTS

Airplane engine jet	Wolf/coyote/dog howling	Car	Animal hiss	Morse code	Typing	Male voice
Baby whining or crying	Telephone ringing digital	Bear	Bell electric	Frog	Saw manual	Rattlesnake
Bell mechanic	Non-vocal music	Big cat	Crowd applause	Music	Thunder	Insect buzz
Bite chew eat	Noise background	Bus	Buzzer	Speech	Pig	Horse walking
Airplane engine propeller	Voice with background noise	Cat meowing	Donkey	Vocal music	Helicopter	Train
Child voice	Telephone ringing bell	Cow	Child laughing	Paper	Saw electric	Hammering
Clean background	Telephone band	Birds	Wind	Sheep	Gun shot heavy	Water
Digital beep	Voice with background music	Dog barking	Dolphin	Siren	People talking	Glass
Chicken clucking	Walk/run/climb stairs (soft)	Female voice	Drink	Whistle	Fireworks	Traffic
Elephant or trumpet	Walk/run/climb stairs (hard)	Electricity	Explosion	Motorcycle	Insect chirp	Fire
Door open or close	Gun shot light	Horn vehicle	Music background		Moose or elk or deer	

and gradual transition detection, respectively) to these datasets resulted in 3459 and 6665 shots; manual grouping of them to scenes resulted in 525 and 357 ground truth scenes. For each of these two datasets, one additional video of the same genre (one documentary, one movie) was processed in the same way (shot segmentation, manual grouping of the shots to scenes) and was used for automatically adjusting the parameters of the algorithm [weights w_y and threshold T in (3) and (4), as well as optimal number of employed visual concept and audio event detectors] in the relevant reported experiments. The third, smaller, dataset was generated with the purpose of simulating unedited news video content; this was done by concatenating several news-related videos from YouTube into three 1-h-long videos. The number of automatically detected shots and manually identified ground truth scenes in the latter dataset was 1763 and 57, respectively.

For evaluating the results of the scene segmentation experiments, the Coverage (C), Overflow (O), and F-Score (F) measures were employed. Coverage and Overflow were proposed in [38] for scene segmentation evaluation; Coverage measures to what extent frames belonging to the same scene are correctly grouped together, while Overflow evaluates the quantity of frames that, although not belonging to the same scene, are erroneously grouped together. The optimal values for Coverage and Overflow are 100% and 0%, respectively. The F-Score is defined in this paper as the harmonic mean of C and $1-O$, to combine Coverage and Overflow in a single measure, $F=\frac{2C(1-O)}{C+(1-O)}$, where $1-O$ is used in this formula instead of O to account for 0 being the optimal value of the latter, instead of 1.

B. Experimental Upper Bounds of Performance

A first series of experiments was carried out with the GSTG method, using those GSTG parameter values that were determined by exhaustive search as being the ones that maximize the F-Score attained for each test dataset. This was done for experimentally estimating an upper bound for the performance of GSTG when different audiovisual features or combinations of them are used. It is reminded that parameters of the GSTG method are the weights w_y and threshold T in (3) and (4); the number of employed visual concept and audio event detectors, assuming that we consider the possibility of using just a subset of those defined in Section V, is also treated as a parameter in this series of experiments. In this and all subsequent series of experiments, in any case where the use of keyframes was required, three keyframes per shot were used. The number $\mathcal{P}$ of STGs of each type that were constructed using randomly selected parameters τ and D_t was set to 1000, with the randomly selected values of τ being in the range $[0, 5000]$ (measured in frames) and of D_t in the range $[0, 0.2]$ or $[0, 0.4]$, depending on the type of STGs. Random selection was implemented with the use of simple random number generators.

The results of GSTG are shown in Table III. The first column ("Index y") indicates the types of STGs that contribute to GSTG in each experiment. The Coverage, Overflow, and F-Score columns report the results of GSTG when the algorithm of [5] is used for individual STG construction, while the F-Score values in parentheses correspond to the case where the fast approximation of Section III is used instead, as part of GSTG. In the first experiment, e.g., $y \in \{V\}$ indicates that only the typical visual features of Section V-A are employed; thus, the resulting method essentially resembles the original STG method of [5], integrating however the probabilistic technique introduced in Section IV-B that alleviates the need for experimentally setting STG construction parameters τ, D_t. In subsequent experiments of this series, STGs constructed with the use of visual concept-based model vectors (VC), typical audio features (A), and audio event-based model vectors (AE), as well as combinations of them, contribute to GSTG. It can be seen from this table that, among individual features (first four rows of the table), the use of the typical visual features results in the highest F-Score. Considering the cases where two or more types of STGs contribute to GSTG, however, it is clear that the $\{V, VC\}$ combination performs better than $\{V\}$ and the $\{A, AE\}$ combination performs better than $\{A\}$. Further combining visual and audio features ($y \in \{V, VC, A\}$ and $y \in \{V, VC, A, AE\}$) leads to additional gains; the F-Score attained by the GSTG when all audiovisual features of Section V are employed is about ten points higher that that of $y \in \{V\}$. The conclusion here is that, providing that good GSTG parameter values can be determined, the GSTG can effectively use any single one of the considered audiovisual features toward improved performance, and the observed performance improvements are significant in both examined datasets. Furthermore, the use of the fast approximation of Section III instead of [5], as part of GSTG, results in only small F-Score degradation (in most cases, F-Score differences

TABLE III
GSTG PERFORMANCE, USING GSTG PARAMETER VALUES THAT WERE DETERMINED BY EXHAUSTIVE SEARCH AS BEING THE ONES THAT MAXIMIZE THE F-SCORE ATTAINED FOR EACH TEST DATASET

	Documentary Dataset			Movie Dataset		
Index y (Types of STGs in GSTG)	Coverage (%)	Overflow (%)	F-Score (%)	Coverage (%)	Overflow (%)	F-Score (%)
$\{V\}$	78.33	19.06	79.61 (78.17)	74.49	24.11	75.18 (74.21)
$\{VC\}$	75.66	31.19	72.07 (71.21)	65.78	17.73	73.11 (71.63)
$\{A\}$	68.58	27.59	70.44 (70.63)	62.33	45.51	58.15 (57.40)
$\{AE\}$	72.24	34.78	68.55 (68.75)	60.28	37.21	61.51 (61.42)
$\{V, VC\}$	80.60	14.71	82.91 (81.57)	71.96	**8.51**	80.56 (80.32)
$\{A, AE\}$	70.10	15.46	76.65 (75.97)	66.16	32.78	66.69 (66.12)
$\{V, VC, A\}$	85.48	12.28	86.59 (86.42)	81.89	15.60	83.13 (83.47)
$\{V, VC, A, AE\}$	**87.35**	**9.37**	**88.96 (88.34)**	**89.27**	17.02	**86.01 (85.55)**

of $< 1\%$) in return for major computational efficiency gains (Section VI-F). These F-Score differences translate to an increase of the number of true scene boundaries that are not detected by less than 1%.

C. Impact of Parameters on Performance

Having examined the performance of GSTG when using "good" GSTG parameter values, we then examined the impact of each of these parameters separately. Starting with the number J_v of visual concept detectors that are taken into account (5), experiments were carried out with it varying from 10 to 90 with a step of 10; the use of all 101 visual concept detectors was also examined. Assuming that, when selecting a subset of the available detectors, it makes sense to select the best J_v detectors out of all the available ones, two different "goodness" criteria were used for the detectors: average precision (AP) and delta average precision (ΔAP) [39]. Both AP and ΔAP for the trained concept detectors were those calculated on the test portion of the TRECVID 2005 dataset (Section V-B). The results presented in Fig. 6 indicate that when $y \in \{VC\}$, higher J_v values generally lead to higher F-Score. When considering combinations of features, though, J_v values between 40 and 80 lead to the best results, using additional concept detectors leads to slight performance decrease. A possible explanation of this is that even poorly performing concept detectors tend to produce similar responses for "similar" keyframes (if not semantically similar, at least visually similar). Thus, in the absence of other features, such concept detectors provide some useful information to the scene boundary detection algorithm, besides introducing noise due to their poor performance in detecting specific concepts. When used in combination with other features, though (specifically, low-level visual features), visual similarity can be reliably estimated from the latter features, and the poorly performing concept detectors seem to only introduce additional noise to the representation of the shots. This noise is responsible for the slight decline of the F-Score when increasing the value of J_v beyond an optimal one. In the above cases, selecting the detectors according to ΔAP is advantageous, compared to using AP, although the differences between the two are generally small ($< 1\%$ in F-Score). What is most interesting though is that regardless of the value of J_v, $y \in \{V, VC\}$ consistently performs better than the baseline $y \in \{V\}$. Furthermore, when additional features are introduced ($y \in \{V, VC, A\}$, $y \in \{V, VC, A, AE\}$), the F-Score curves as a function of J_v tend to become more flat, i.e., although $\{VC\}$ introduces significant performance improvement (particularly for the Movie dataset), GSTG is rather insensitive to the number of employed visual concept detectors.

A similar study of the number J_a of employed audio event detectors was also carried out, with J_a ranging from 20 to 60 with a step of 10; using all 75 audio events of Table II was also examined. The F-Score of each individual audio event detector, calculated on the test portion of the audio event corpus (Section V-D), was used as a detector goodness criterion. The results, shown in Fig. 7, are similar to those for the visual concept detectors that were discussed above.

Finally, regarding the impact of weights w_y and threshold T when $y \in \{V, VC, A, AE\}$), results from varying each of w_{VC}, w_A, w_{AE}, and T separately are shown in Fig. 8. In varying the weights, w_V was set equal to $1 - w_{VC} - w_A - w_{AE}$; thus, in Fig. 8(a), w_{VC} varies from 0 to 100% of its maximum allowed value, the latter being the one that would make w_V equal to 0 for the given (constant) values of w_A and w_{AE}, similarly for w_A and w_{AE} in Fig. 8(b) and (c), respectively. The results indicate that GSTG is not very sensitive to the values of weights w_y, since in all cases there is a relatively large range of weight values that result in close-to-maximum F-Score, and no abrupt changes in F-Score for small changes in a weight value are observed. Threshold T [Fig. 8(d)] is shown to have a more significant impact on F-Score, which was however expected, considering that its minimum and maximum values practically mean that all and no potential scene boundaries, respectively, are accepted as scene boundaries. Even for T, though, there is a relatively large range of values that result in close-to-maximum F-Score.

D. Results Using Automatically Determined Parameters and Comparison with Literature Works

An advantage of the GSTG approach, discussed in Section IV-B, is that weights w_y of GSTG are not hard-to-optimize STG construction parameters; on the contrary, they can be easily optimized using LSE. In this section, we repeat the series of experiments of Section VI-B, using however the single out-of-testset video for each dataset that was mentioned in Section VI-A in order to automatically select the values of weights w_y as well as all other GSTG parameters (T, J_v,

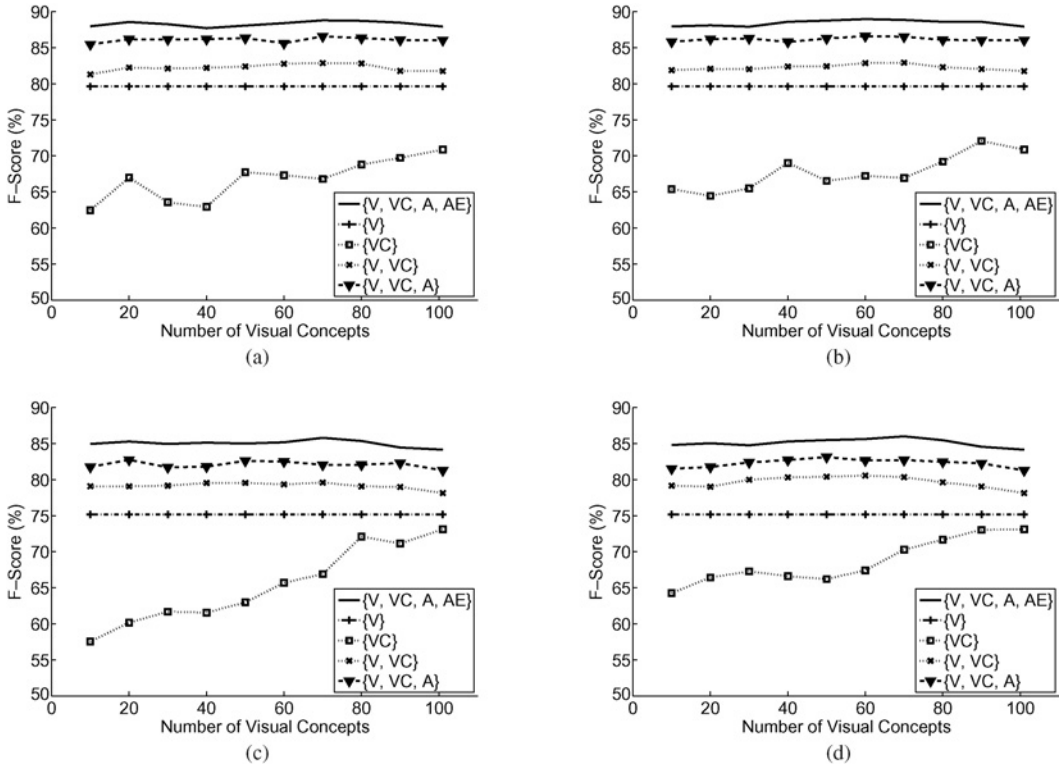

Fig. 6. F-Score as a function of the number J_v of visual concept detectors. (a) Documentary dataset, concepts selected according to AP. (b) Documentary dataset, concepts selected according to ΔAP. (c) Movie dataset, concepts selected according to AP. (d) Movie dataset, concepts selected according to ΔAP.

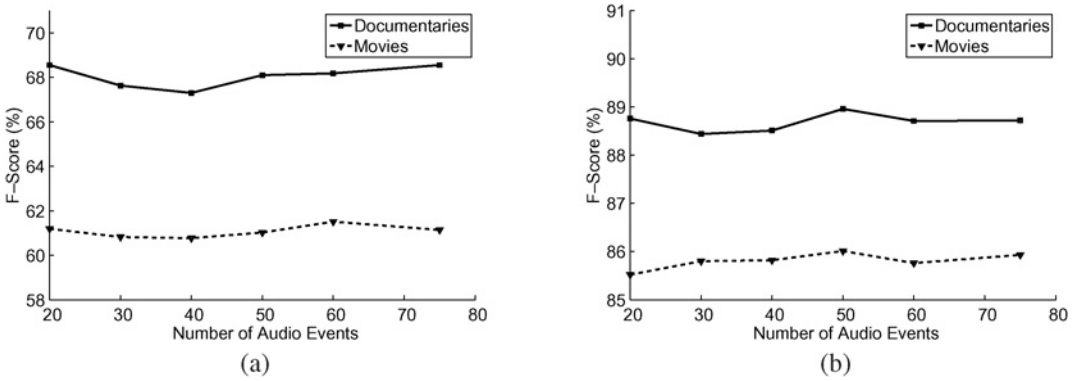

Fig. 7. F-Score as a function of the number J_a of audio event detectors. (a) $y \in \{AE\}$. (b) $y \in \{V, VC, A, AE\}$.

J_a). For weights w_y, LSE estimation is employed. Specifically, a value of 1 is assigned to each shot boundary of the out-of-testset ground-truth-segmented video that is also a scene boundary, according to the ground-truth segmentation, and a value of 0 to each other shot boundary. LSE estimates the weights w_y that minimize the sum of differences between the aforementioned values and p_i (3) for this video. Threshold T is then set to the value that maximizes the F-Score attained for the same out-of-testset video, given the estimated weights; this value is determined by simple exhaustive search. Finally, the above optimization process is repeated for different selected values of J_v and J_a (the same few values used for plotting Figs. 6 and 7), and the set of parameters that leads to the maximum F-Score for the out-of-testset video is chosen. Although this may not be the most elegant optimization process possible, it is a simple one that requires use of just one out-of-testset ground-truth-segmented video for automatically estimating all parameters of GSTG. The results of using the estimated parameters on the test datasets are reported in the first part of Table IV. Again, the Coverage, Overflow, and F-Score columns report the results of GSTG when the algorithm of [5] is used for individual STG construction, while the F-Score values in parentheses correspond to the case where the fast approximation of Section III is used instead, as part of GSTG. It can be seen that, in comparison to the results of Table III, the F-Score in almost all experiments has only been slightly reduced (F-Score differences of approximately 1%). The F-Score attained by the GSTG when all audiovisual features of Section V are employed continues to be about ten points higher that that of $y \in \{V\}$, and every one of the 4 examined types of features is shown to have a non-negligible contribution. The conclusion here is that automatic selection

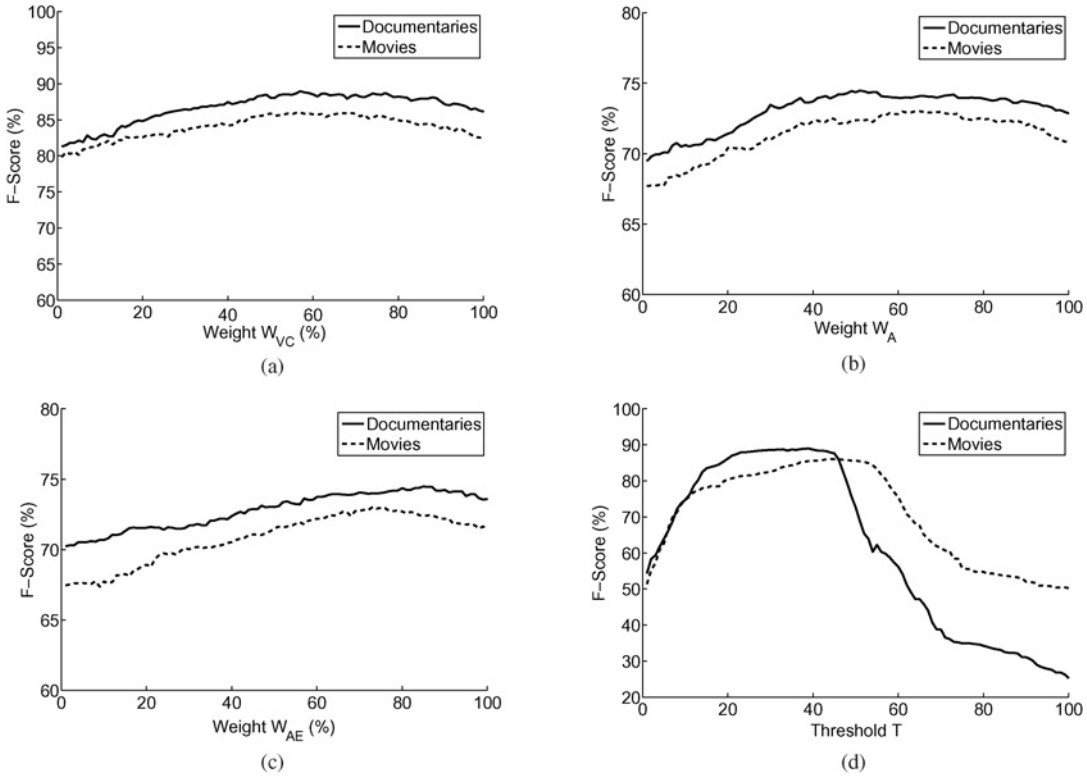

Fig. 8. F-Score as a function of weights w_y and threshold T. (a) F-Score versus w_{VC}. (b) F-Score versus w_A. (c) F-Score versus w_{AE}. (d) F-Score versus T.

of GSTG parameter values using a simple procedure and a single out-of-testset video of the same genre is sufficient for getting very close to the upper performance bounds identified in Section VI-B.

For the purpose of comparing the proposed GSTG method with additional methods of the literature, besides the STG [5] (whose results, when it also exploits the probabilistic technique introduced in Section IV-B, are essentially those reported above for $y \in \{V\}$), three additional methods are tested and their results are also reported in Table IV. These are the very recent unimodal method of [12], which is based on an elaborate sequence alignment technique, and the multimodal methods of [21] and [24], which similarly to GSTG combine visual and audio features. The latter method [24] is based on a discriminative classifier (SVM) that realizes early fusion of the audio-visual features. For ensuring a fair comparison, the same keyframes, audio segmentation results, and high-level audio features (where applicable) that are used by the proposed approach were also used when experimenting with these three methods. It can be seen from the reported results that the GSTG method significantly outperforms [12], [21], and [24]. These performance differences are caused by the use of a wealth of low-level and high-level audiovisual features in the proposed approach, as opposed to just low-level features being used in [12] and [21]. The proposed probabilistic merging process that effectively combines these features also contributes to improved performance, in comparison to simpler heuristics used in [21] for audiovisual feature combination, and also in comparison to early fusion of low-level and high-level audiovisual features used in [24].

Finally, in the last row of Table IV, results of the GSTG are reported for the case that the weights w_y and all other GSTG parameters are automatically selected with the use of an out-of-testset ground-truth-segmented video that belongs to a different genre (i.e., one documentary video is used for estimating the GSTG parameters for the movie dataset, and similarly one movie video is used for the documentary dataset). For both datasets, this cross-genre automatic parameter selection results in F-Score differences of $< 0.5\%$, compared to using a same-genre video for this task. These results complement our previous findings about the insensitivity of the proposed technique to parameters (Section VI-C), and indicate that the GSTG can in practice be applied to different video genres without using even one manually segmented video of the same genre, with minimal performance loss.

E. Applicability of GSTG to News Videos

In order to discuss the applicability of the GSTG approach to different video genres, most notably news-related videos, we first need to make the distinction between two broad types of video content: loosely structured content and tightly structured one. We use the term "tightly structured content" here to denote content that is known to follow a very specific structure. Examples of such video are the news bulletins of a single broadcaster; they tend to follow a structure that is characteristic of the broadcaster, e.g., each scene starts with one anchor-person shot and is followed by external reporting shots. On the contrary, video genres such as documentaries, movies, unedited news-related video, and others, do not observe such strict structuring rules, and consequently fall

TABLE IV
GSTG PERFORMANCE, USING GSTG PARAMETER VALUES THAT WERE AUTOMATICALLY ESTIMATED WITH THE USE OF AN OUT-OF-TESTSET GROUND-TRUTH-SEGMENTED VIDEO, AND COMPARISON WITH LITERATURE WORKS [12], [21], [24]

	Documentary Dataset			Movie Dataset		
Index y (Types of STGs in GSTG)	Coverage (%)	Overflow (%)	F-Score (%)	Coverage (%)	Overflow (%)	F-Score (%)
$\{V\}$	76.96	20.80	78.06 (77.10)	73.55	26.11	73.72 (72.81)
$\{VC\}$	76.37	35.37	70.01 (70.53)	71.20	25.68	72.73 (71.36)
$\{A\}$	68.52	28.44	70.01 (68.50)	59.64	44.79	57.34 (57.31)
$\{AE\}$	63.81	28.47	67.45 (67.47)	62.14	40.97	60.55 (60.68)
$\{V, VC\}$	83.29	18.42	82.43 (81.32)	80.62	20.93	79.84 (80.30)
$\{A, AE\}$	70.96	21.18	74.68 (74.41)	66.49	34.42	66.03 (65.18)
$\{V, VC, A\}$	85.44	16.77	84.32 (84.71)	84.77	19.32	82.67 (81.70)
$\{V, VC, A, AE\}$	**86.30**	**10.91**	**87.67 (87.40)**	**87.91**	**17.89**	**84.91 (84.64)**

Method	Coverage (%)	Overflow (%)	F-Score (%)	Coverage (%)	Overflow (%)	F-Score (%)
GSTG ($y \in \{V, VC, A, AE\}$)	**86.30**	**10.91**	**87.67 (87.40)**	**87.91**	17.89	**84.91 (84.64)**
Method of [12]	70.90	24.13	73.30	76.43	**16.15**	79.97
Method of [21]	77.59	17.31	80.06	75.12	24.29	75.41
Method of [24]	78.22	16.73	80.67	79.50	21.17	79.16

GSTG ($y \in \{V, VC, A, AE\}$) + Cross-genre parameter selection	Coverage (%)	Overflow (%)	F-Score (%)	Coverage (%)	Overflow (%)	F-Score (%)
	85.93	**11.40**	**87.24 (87.22)**	**87.52**	**18.17**	**84.58 (84.37)**

under the category of loosely structured content. In the case of tightly structured content, it is evidently advantageous to develop dedicated methods that exploit the knowledge about the content's structure (thus focusing, e.g., on detecting the anchor-person shots that may signify a scene change). The GSTG approach, on the contrary, similarly to most literature works, is a generic approach that does not make any restrictive assumptions about the structure of the video, thus is mostly suited for processing loosely structured content.

For examining how the GSTG performs on news-related content falling under the latter category, we used the third dataset defined in Section VI-A, which simulates unedited news video content. Application of GSTG to it (with the fast STG approximation of Section III being used as part of GSTG: $y \in \{V, VC, A, AE\}$) and looking for the experimental upper bounds of performance (as in Section VI-B) resulted in F-Score equal to 78.76%; automatically determining the GSTG's parameters resulted in F-Scores equal to 77.91% and 77.83%, when a documentary and a movie were used for cross-genre parameter selection, respectively (as in the last paragraph of Section VI-D). In comparison, the F-Scores for the literature works [12], [21], [24] were 75.97%, 75.09%, and 75.19%, respectively.

F. Computational Efficiency

Concerning the computational efficiency of the GSTG approach, this is experimentally shown to be high. Specifically, excluding the pre-processing of the audio-visual stream (e.g., shot segmentation) and feature extraction, the GSTG approach is faster than real-time (approximately 60 f/s) on an 3.0 GHz personal computer, considering $y \in \{V, VC, A, AE\}$ and employing the method of [5] for individual STG construction. When, instead of the latter, the fast STG approximation introduced in this paper is used as part of GSTG, the frame processing rate rises to over 1200 f/s, representing a speed-up by over 20 times. As a result, the processing time for a 90-min film (featuring 25 f/s) is reduced from about 40 min to less than 2. The pre-processing and feature extraction processes excluded from the aforementioned time measurements clearly introduce some additional computational overhead; nevertheless: 1) some of these processes (e.g., shot segmentation) are common to all scene segmentation methods; 2) other processes (e.g., concept detection) are typically performed on the video as part of its semantic analysis, and re-using their results also for the purpose of scene segmentation does not introduce additional computational cost; and 3) real-time or near-real-time implementations for all of them generally exist (even for concept detection, e.g., [40]).

VII. CONCLUSION

In this paper, a novel multimodal scene segmentation method, making use of high-level audiovisual features, was presented. As part of this method, algorithms were developed: 1) for a fast STG approximation; 2) for extending the STG so as to exploit non-visual input; and 3) for effectively combining STGs that were constructed with the use of different features, possibly coming from processing different modalities of the audiovisual stream. New high-level features, such as model vectors constructed with the help of large numbers of trained visual concept detectors or audio event detectors, were presented and were exploited by the proposed multimodal scene segmentation method. For training these detectors, existing annotated corpora were employed; these were unrelated to the datasets used for experimentation in this paper, thus not leaving room for any doubts on the usefulness of the model vector-based features on different datasets. The experimental results revealed the merit of the developed algorithms and documented the significance of introducing high-level audiovisual features in the scene segmentation task.

REFERENCES

[1] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello, "Foveated shot detection for video segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 365–377, Nov. 2005.

[2] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 580–588, Jun. 1999.

[3] R. Jain, "Eventweb: Developing a human-centered computing system," *IEEE Comput.*, vol. 41, no. 2, pp. 42–50, Feb. 2008.

[4] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of trecvid activity," *Comput. Vision Image Understand.*, vol. 114, no. 4, pp. 411–418, Apr. 2010.

[5] M. Yeung, B.-L. Yeo, and B. Liu, "Segmentation of video by clustering and graph analysis," *Comput. Vision Image Understand.*, vol. 71, no. 1, pp. 94–109, 1998.

[6] W. Tavanapong and J. Zhou, "Shot clustering techniques for story browsing," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 517–527, Aug. 2004.

[7] Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1097–1105, Dec. 2005.

[8] Y. Zhai and M. Shah, "Video scene segmentation using Markov chain Monte Carlo," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 686–697, Aug. 2006.

[9] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 296–305, Feb. 2005.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[11] Y. Zhao, T. Wang, P. Wang, W. Hu, Y. Du, Y. Zhang, and G. Xu, "Scene segmentation and categorization using NCuts," in *Proc. CVPR*, 2007, pp. 1–7.

[12] V. T. Chasanis, A. C. Likas, and N. P. Galatsanos, "Scene detection in videos using shot clustering and sequence alignment," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 89–100, Jan. 2009.

[13] Y. Cao, W. Tavanapong, K. Kim, and J. Oh, "Audio-assisted scene segmentation for story browsing," in *Proc. ACM CIVR*, 2003, pp. 446–455.

[14] Y. Zhu and D. Zhou, "Scene change detection based on audio and video content analysis," in *Proc. 5th Int. Conf. Computat. Intell. Multimedia Applicat.*, 2003, pp. 229–234.

[15] H. Sundaram and S.-F. Chang, "Video scene segmentation using video and audio features," in *Proc. IEEE ICME*, Jul.–Aug. 2000, pp. 1145–1148.

[16] S. Rho and E. Hwang, "Video scene determination using audiovisual data analysis," in *Proc. 24th Int. Conf. Distribut. Comput. Syst. Workshops*, 2004, pp. 124–129.

[17] S.-C. Chen, M.-L. Shyu, W. Liao, and C. Zhang, "Scene change detection by audio and video clues," in *Proc. IEEE ICME*, Nov. 2002, pp. 365–368.

[18] A. Chianese, V. Moscato, A. Penta, and A. Picariello, "Scene detection using visual and audio attention," in *Proc. Ambi-Sys Workshop Ambient Media Delivery Interact. Television*, 2008, pp. 4030–4033.

[19] S. Pfeiffer, R. Lienhart, and W. Efflsberg, "Scene determination based on video and audio features," *Multimedia Tools Applicat.*, vol. 15, no. 1, pp. 59–81, 2001.

[20] A. Velivelli, C.-W. Ngo, and T. S. Huang, "Detection of documentary scene changes by audio-visual fusion," in *Proc. ACM CIVR*, 2004, pp. 227–237.

[21] N. Nitanda, M. Haseyama, and H. Kitajima, "Audio signal segmentation and classification for scene-cut detection," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4. May 2005, pp. 4030–4033.

[22] V. Parshin, A. Paradzinets, and L. Chen, "Multimodal data fusion for video scene segmentation," in *Proc. Int. Conf. Vis. Inform. Inform. Syst.*, 2005, pp. 279–289.

[23] N. Goela, K. Wilson, F. Niu, and A. Divakaran, "An SVM framework for genre-independent scene change detection," in *Proc. IEEE ICME*, Jul. 2007, pp. 532–535.

[24] K. Wilson and A. Divakaran, "Discriminative genre-independent audio-visual scene change detection," *Proc. SPIE Conf. Multimedia Content Access: Algorithms Syst. III*, vol. 7255, p. 725502, Jan. 2009.

[25] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, and I. Trancoso, "Multi-modal scene segmentation using scene transition graphs," in *Proc. ACM Multimedia*, 2009, pp. 665–668.

[26] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, "On the use of audio events for improving video scene segmentation," in *Proc. WIAMIS*, Apr. 2010, pp. 1–4.

[27] J. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *Proc. IEEE ICME*, Jul. 2003, pp. 445–448.

[28] N. Rasiwasia, P. Moreno, and N. Vasconcelos, "Bridging the gap: Query by semantic example," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 923–938, Aug. 2007.

[29] C. Snoek, M. Worring, J. van Gemert, J.-M. Geusebroek, and A. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proc. ACM Multimedia*, Oct. 2006, pp. 421–430.

[30] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 703–715, Jun. 2001.

[31] V. Mezaris, P. Sidiropoulos, A. Dimou, and I. Kompatsiaris, "On the use of visual soft semantics for video temporal decomposition to scenes," in *Proc. IEEE ICSC*, Sep. 2010, pp. 141–148.

[32] R. Amaral, H. Meinedo, D. Caseiro, I. Trancoso, and J. Neto, "A prototype system for selective dissemination of broadcast news in European portuguese," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 37507, pp. 1–11, 2007.

[33] H. Meinedo, "Audio pre-processing and speech recognition for broadcast news," Ph.D. thesis, IST, Tech. Univ. Lisbon, Lisbon, Portugal, Mar. 2008.

[34] M. Bugalho, J. Portelo, I. Trancoso, T. Pellegrini, and A. Abad, "Detecting audio events for semantic video search," in *Proc. Interspeech*, Sep. 2009, pp. 1151–1154.

[35] I. Trancoso, T. Pellegrini, J. Portelo, H. Meinedo, M. Bugalho, A. Abad, and J. Neto, "Audio contributions to semantic video search," in *Proc. IEEE ICME*, Jul. 2009, pp. 630–633.

[36] G. Chavez, M. Cord, S. Philip-Foliguet, F. Precioso, and A. Araujo, "Robust scene cut detection by supervised learning," in *Proc. EUSIPCO*, Sep. 2006.

[37] E. Tsamoura, V. Mezaris, and I. Kompatsiaris, "Gradual transition detection using color coherence and other criteria in a video shot meta-segmentation framework," in *Proc. IEEE ICIP-MIR*, Oct. 2008, pp. 45–48.

[38] J. Vendrig and M. Worring, "Systematic evaluation of logical story unit segmentation," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 492–499, Dec. 2002.

[39] J. Yang and A. Hauptmann, "(Un)reliability of video concept detection," in *Proc. ACM CIVR*, Jul. 2008, pp. 85–94.

[40] J. Uijlings, A. Smeulders, and R. Scha, "Real-time bag of words, approximately," in *Proc. ACM CIVR*, 2009.

Panagiotis Sidiropoulos received the Diploma degree in electrical and computer engineering in 2003 and the M.S. degree in informatics in 2007, both from the Aristotle University of Thessaloniki, Thessaloniki, Greece. Since January 2010, he has been pursuing the Ph.D. degree under Prof. J. Kittler, University of Surrey, Guildford, Surrey, U.K.

In June 2008, he joined CERTH/ITI, Thessaloniki, as a Research Associate. His current research interests include the fields of multimedia retrieval, image and video analysis, and machine learning.

Vasileios Mezaris (S'98–M'06) received the Diploma and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2005, respectively.

He is currently a Senior Researcher (Researcher C) with the Informatics and Telematics Institute/Center for Research and Technology Hellas, Thermi-Thessaloniki. His current research interests include image and video analysis, content-based and semantic image and video retrieval, event detection in multimedia, and medical image analysis. He is the co-author of 19 papers in refereed international journals, 6 book chapters, and more than 60 papers in international conferences.

Ioannis Kompatsiaris (S'94–M'02–SM'11) received the Ph.D. degree in 3-D model-based image sequence coding from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001.

He is currently a Senior Researcher (Researcher B) with the Informatics and Telematics Institute, Thermi-Thessaloniki. His current research interests include semantic multimedia analysis, indexing and retrieval, social media analysis, knowledge structures, and reasoning and personalization for multimedia applications. He is the co-author of 42 papers in refereed journals, 20 book chapters, 4 patents, and more than 150 papers in international conferences.

Dr. Kompatsiaris has been the co-organizer of various international conferences and workshops and has served as a regular reviewer for a number of journals and conferences. He is a member of ACM.

Hugo Meinedo received the Licenciado, Mestre, and Doutor (Ph.D.) degrees in electrical and computer engineering from the Instituto Superior Tecnico, Lisbon, Portugal, in 1996, 2000, and 2008, respectively. His Ph.D. thesis focused on audio pre-processing and speech recognition for broadcast news.

He joined the Neural Networks and Signal Processing Group of INESC, Lisbon, in 1997. Since 2001, he has been a Researcher with the Spoken Language Systems Laboratory (L2F), INESC-ID, Lisbon. He has participated in several European and national projects. His current research interests include neural networks, audio processing, real-time speech recognition algorithms, and multimodal audio-visual applications.

Miguel Bugalho received the Licenciado, Mestre, and Doutor (Ph.D.) degrees in information systems and computer engineering from the Instituto Superior Tecnico, Lisbon, Portugal, in 2002, 2004, and 2010, respectively. His Ph.D. thesis focused on machine learning and search techniques for bioinformatics.

He joined the Algorithms for Optimization and Simulation Group in 2001 followed by the Knowledge Discovery and Bioinformatics Group in 2006, both of INESC, Lisbon. Since 2008, he has been a Researcher with the Spoken Language Systems Laboratory (L2F), INESC-ID, Lisbon. He has participated in several European and national projects. His current research interests include machine learning techniques applied to audio processing, audio-event detection, sentiment analysis, and multimodal audio-visual applications.

Isabel Trancoso (SM'03–F'11) is a Full Professor with the Instituto Superior Tecnico and a Senior Researcher at INESC-ID, Lisbon, Portugal, having launched the Speech Processing Group, now restructured as L2F, in 1990. Her first research topic was medium-to-low bit rate speech coding. From October 1984 to June 1985, she worked on this topic at AT&T Bell Laboratories, Murray Hill, NJ. Her current scope is much broader, encompassing many areas in speech recognition and synthesis, with a special emphasis on tools and resources for the Portuguese language. Her expertise has been internationally recognized by leading institutions in the spoken language processing area. She is currently the President of the International Speech Communication Association.

She has participated in the Signal Processing Society Board of Governors, and acted as Editor-in-Chief of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING.