A Multimodal Scheme for Program Segmentation and Representation in Broadcast Video Streams

Jinqiao Wang, Lingyu Duan, Qingshan Liu, *Member, IEEE*, Hanqing Lu, *Member, IEEE*, and Jesse S. Jin, *Member, IEEE*

***Abstract*—With the advance of digital video recording and playback systems, the request for efficiently managing recorded TV video programs is evident so that users can readily locate and browse their favorite programs. In this paper, we propose a multimodal scheme to segment and represent TV video streams. The scheme aims to recover the temporal and structural characteristics of TV programs with visual, auditory, and textual information. In terms of visual cues, we develop a novel concept named program-oriented informative images (POIM) to identify the candidate points correlated with the boundaries of individual programs. For audio cues, a multiscale Kullback–Leibler (K–L) distance is proposed to locate audio scene changes (ASC), and accordingly ASC is aligned with video scene changes to represent candidate boundaries of programs. In addition, latent semantic analysis (LSA) is adopted to calculate the textual content similarity (TCS) between shots to model the inter-program similarity and intra-program dissimilarity in terms of speech content. Finally, we fuse the multimodal features of POIM, ASC, and TCS to detect the boundaries of programs including individual commercials (spots). Towards effective program guide and attracting content browsing, we propose a multimodal representation of individual programs by using POIM images, key frames, and textual keywords in a summarization manner. Extensive experiments are carried out over an open benchmarking dataset TRECVID 2005 corpus and promising results have been achieved. Compared with the electronic program guide (EPG), our solution provides a more generic approach to determine the exact boundaries of diverse TV programs even including dramatic spots.**

***Index Terms*—Broadcast video, latent semantic analysis, multimodal fusion, TV program segmentation.**

I. Introduction

TV is "central to the entertainment, information, leisure, and social life of millions of homes all over the world" [1]. With ever-increasing TV channels, we are exposed to overwhelming amounts of TV programs. However, it is almost always impossible to catch all of our favorite shows directly from TV broadcasting due to time conflict. Although personal video recorders (PVR) like Tivo [2] have extended the capability of TV viewers, it is really time consuming to search a special program from a large-scale raw video archive. Information overload often makes people "lost in TV program space" [3]. Consequently, to efficiently manipulate and manage TV programs is significant. Most relevant work [4]–[9] has focused on video scene detection and classification from the perspective of video retrieval. Undoubtedly, the diversity of video scenes could limit the scene-based approaches to TV program management. As the definition of scenes is ambiguous by nature, it is extremely difficult, if not impossible, to provide a generic solution covering most user interests by predetermining considerable categories of scenes. In this paper, we present a novel program-oriented solution to manipulate and manage TV video streams. Different from the scene-based approaches [10], our solution makes use of program-level broadcast video production knowledge, which is characterized by explicit structural information and more rich program-oriented semantics that enable users to conveniently search their favorite programs in practice. Towards the program-level manipulation and management of TV streams, we have to address two subproblems: 1) how to accurately segment individual programs and 2) how to properly represent programs for user browse and search.

A. Challenges to Structuring TV Streams

Scene detection and classification is a traditional approach to the manipulation and management of video data. In TV streams, the broadcasting programs involving different genres or even the programs of the same genre always exhibit fairly diverse scenes. Scene is in essence a subjective concept. Such diverse TV programs render it difficult to make a clear and meaningful definition for scenes in general. In practice, a computational scene is resolved by evaluating the spatiotemporal similarity in terms of the low-level audiovisual features amongst video shots. From the computation point of view, different user semantics are ideally associated with distinct clustering results of shots or scenes in video content analysis. Unfortunately, it is extremely difficult to accurately locate the definite boundaries between TV programs through a shot or scene based clustering approach. As a program comprises a number of scenes and stories, the problem of program segmentation is different from that of scene or story segmentation. To model the behaviors of shot-level coherence is less effective for delineating the boundaries of programs.

In addition, automatic genre classification is a sort of useful approaches to managing programs. Genre classification often requires *a priori* knowledge (e.g., program boundaries) and semantics modeling. Such approaches are limited in managing

Manuscript received January 21, 2007; revised November 16, 2007. This work was supported by the National Natural Science Foundation of China under Grants 60475010, 60121302, and 60675003, and by the 863 Program 2006AA01Z315. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jie Yang.

J. Wang, Q. Liu, and H. Lu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: jqwang@nlpr.ia.ac.cn; qsliu@nlpr.ia.ac.cn; luhq@nlpr.ia.ac.cn).

L. Duan is with the Institute for Infocomm Research, Singapore 119613 (e-mail: lingyu@i2r.a-star.edu.sg).

J. S. Jin is with the School of Design, Communication, and Information Technology, University of Newcastle, NSW 2308, Australia (e-mail: jesse.jin@newcastle.edu.au).

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/TMM.2008.917362

open and extensive TV streams so that the applicability is challenged. It is well known that even for those programs of the same genre, video content production and program composition arts may vary in different TV channels.

Hence, most existing video structuring approaches cannot elegantly address the problem of program segmentation from TV streams. Fairly various programs, dramatic content changes in lighting, chromatic composition and creative stories have made existing shot-level [11], [12], or scene-level [4]–[9] structuring or genre categorization [13]–[20] methods less effective for program segmentation and subsequent manipulation and management.

B. Our Solution to the Segmentation and Representation of TV Programs

We propose a novel multimodal scheme to recover the temporal structure of TV streams and provide a vivid representation for program guide and browse. Different from scene or story segmentation, program segmentation is basically oriented towards definite semantics, namely, the objective boundaries of programs. Multimodal features such as program oriented informative images (POIM) [21], audio scene change (ASC), and textual content similarity (TCS) are developed to represent the structure characteristics relevant to inter- or intra-TV programs.

A novel visual concept POIM is introduced to capture the apparent visual information relevant to program composition in TV streams. Although diverse programs are presented in TV channels, almost all kinds of programs have a common structural feature, namely, ⟨ opening credits, program content, closing credits ⟩. Opening credits are "shown at the beginning of a show and to list the most important members of the production [22]." The visual representation usually involves the texts superimposed on a blank screen or a still picture, or on top of actions in a show, or sometimes around an animation. Closing credits "come at the end of a show and list all the cast and crew involved in the production [23]." They are usually shown in small characters. These images in the shots of opening credits and closing credits may provide significant information to recover the macro temporal structure of TV streams. For example, the program "DECISION 2004" begins with the shot (opening credit) containing the superimposed text "DECISION 2004" and ends with the shot (closing credit) containing the caption "GEICO" that indicates the production and sponsor information, as shown in Fig. 2.

Since POIM images almost always appear at the beginning or the end of a TV program, they become one of important indicators to segment programs in our solution. In addition, ASC and TCS are meaningful to determine the boundaries. As the transitions between TV programs must occur at shot changes (e.g., cut, or fade in/out). Consequently, the problem of program segmentation is transformed to a binary classification problem of identifying true or false transitions at the candidate points based on the features of POIM, ASC, and TCS. Accurate shot segmentation (especially high recall rate) is significant to identity candidate the points of program transitions.

Clearly the proper modeling of POIM, ASC, and TCS is necessary. We employ support vector machines (SVM) [24] to train the classifier of POIM images. A so-called multiscale Kullback–Leibler (K–L) distance measure [25] is proposed to capture the temporal behaviors of low level audio features followed by an audiovisual alignment process at candidate points. ASC are thus detected by measuring the K–L distances at candidate points. Moreover, latent semantic analysis (LSA) [26] is employed to analyze textual content from automatic speech recognition (ASR) transcripts or machine translated (MT) transcripts, and a cosine similarity measure is utilized to measure the TCS between relevant shots. In order to fuse multimodal features, a linear fusion model is adopted in which different weights are set for individual feature to indicate their importance, respectively.

On the basis of program segmentation, we propose a sort of program representation containing visual and textual summaries. A set of key frames are selected from relevant shots while representative keywords are generated by ranking text words. The combination of visual and textual information contributes to a more comprehensive representation of TV programs. In particular, we extract POIM images as a special kind of key frames to visually provide clear semantics about program details, which enable users to discriminate different types of TV programs. In addition, the representative keywords are useful complements of visual information since users may utilize keywords to search their favorite programs. We argue that such a multimodal representation scheme may become one of useful techniques to manipulate and manage TV streams.

The rest of this paper is organized as follows. Section II reviews relevant work. Section III describes the overall framework of program segmentation. Section IV presents the extraction of multimodal features. Program segmentation involving multimodal fusion is discussed in Section V. Section VI introduces the multimodal TV program representation scheme. Experiment results are given in Section VII. Finally, we conclude this paper in Section VIII.

II. Related Work

In this section, we review relevant work on the segmentation of program, story, and scene from video streams. Multimodal fusion approaches are also briefly discussed.

A. TV Program Segmentation and Categorization

There has been lots of work on the segmentation [10], [14], [27] and categorization [13], [15]–[20], [27] of individual programs and commercials in TV streams. Hauptmann *et al.* [14] exploited multimodal features to segment news stories and commercial breaks. Duan *et al.* [27] attempted to segment individual commercials through developing mid-level multimodal features followed by SVM-based binary classification. Huang *et al.* [10] utilized Hidden Markov Model (HMM) to segment programs, but it failed to distinguish the boundaries of individual commercials. In addition to program segmentation, many researchers have studied the problem of program categorization (or genre classification) over a limited set of predefined program categories such as news, weather, sitcoms, sports, commercial and so on. Multimodal features were widely exploited. Fischer *et al.* [13] categorized programs into news, commercial, cartoon, tennis, and car racing. Huang *et al.* [14] presented a hierarchical method to classify programs into news, basketball, weather, and commercial by detecting the auditory, visual and motion breaks.

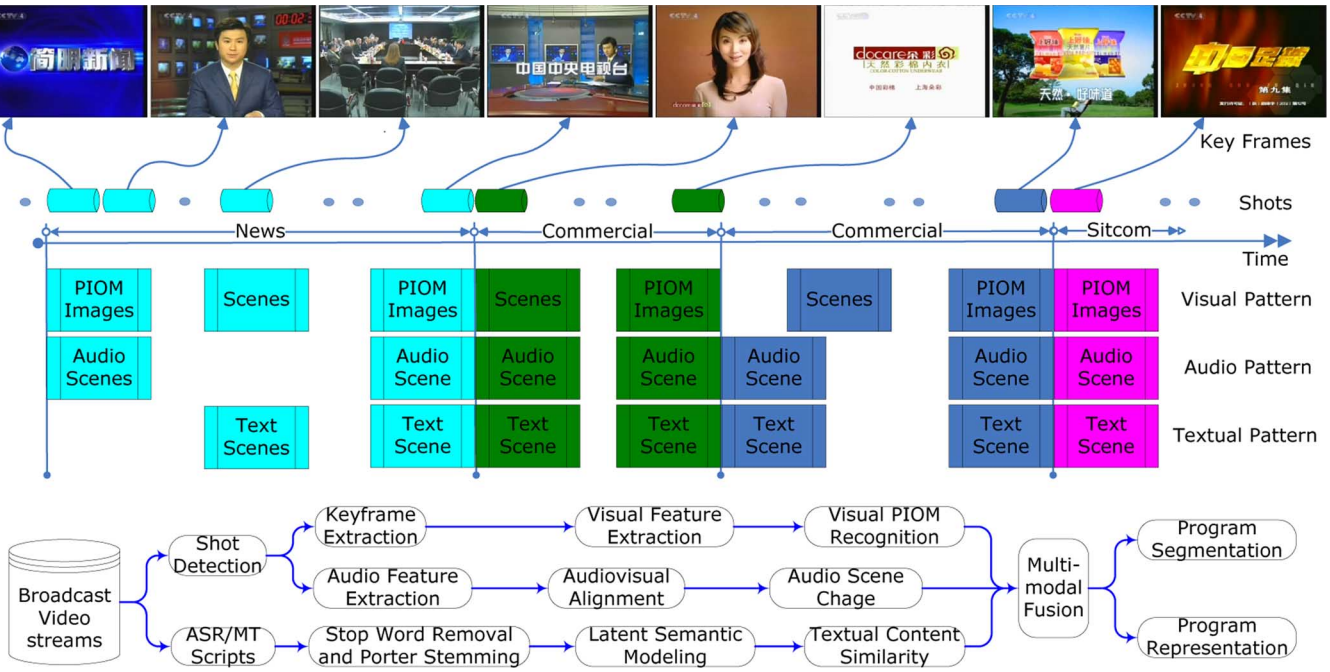


Fig. 1. Overall framework of multimodal program segmentation and representation.

Fig. 2. Examples of POIM images from different TV programs. (From top to bottom: news, commercial, sitcom, and others).

Truong [18] *et al.* combined editing effects, color and motion to classify news, commercial, music video, cartoon, and sports. Gang [19] applied the tracking of face and superimposed text to classify programs into four categories, i.e., news, commercial, sitcom, and soap. Kim [20] tried to distinguish cartoon, commercial, music video, news, sports, and talk show directly from MPEG compressed bit-streams with visual information only. Overall, these program segmentation & categorization approaches focused on the low-level features, in which the use of program production knowledge was de-emphasized. Moreover, the more or less complexity and some unavoidable ad-hoc nature have limited their application in extensive TV streams.

B. Story Segmentation

Generally speaking, story is a basic unit for browsing and summarizing video content of TV programs. Story segmentation is fundamental to video analysis, indexing, and retrieval. The problem of story segmentation has been widely investigated in various kinds of documents, especially in news videos [28]. The problem of story segmentation originates from detecting stories in text documents. In text analysis, a sort of coherence in lexical contents and/or discourse structures in text chunks is detected to identify story boundaries [11]. Text tiling is a basic method wherein a story boundary is declared at the locations having a coherence measure below a threshold [12]. Some variants of this method attempt to examine the coherence in nouns or other semantic entities in texts [29]. Other techniques resort to machine learning approaches like discourse-based approaches [30]. However, the purely text-based methods are constrained as the text distribution statistics alone is often insufficient to detect accurate boundaries in multimedia documents. To improve the accuracy of story detection in multimedia documents (e.g., news video, movies), some researchers have integrated multimodal features, domain-specific produc-

tion rules, and other domain knowledge. For example, [31] utilized a maximum entropy method to fuse 195 features from multiple modalities (audio, visual, and text) to discover stories over news videos, and had achieved a promising accuracy of some 70%. It is worth noting that text is significant even in video story segmentation. A TV program involves a number of stories that are featured by much more relevant texts between stories. Therefore, in addition to audiovisual features, the proper use of textual information can model the dependency or independency between stories for better segmentation of TV programs.

C. Scene Segmentation

Scene segmentation is to group successive shots into meaningful clusters along the temporal dimension, so that all the shots in a scene are relevant to a common topic, which could be on a particular physical setting, an ongoing event, or a theme. Various clustering methods and visual features were proposed to detect scenes. Hanjalic *et al.* [32] tried to detect the boundaries of logical story units in movies by calculating the inter-shot visual similarity based on key frames. Similar shots are linked, and the story segmentation is completed by connecting the overlapping links. Ngo *et al.* [5] integrated both tensor and color histograms to carry out a two-level hierarchical clustering of shots based on temporal slices. Likewise, Okamoto *et al.* [6] used spatio-temporal image slices of fixed length for clustering. Moreover, some advanced statistical models have been proposed to detect scenes. Zhang *et al.* [7] tried to represent videos with K-partite graphs, and applied the spectral clustering to segment stories. Yeung *et al.* [4] proposed a scene transition graph to incorporate temporal constraints for determining story units. Rasheed and Shah [8] constructed an undirected graph according to the shot similarity based on color and motion cues, and used normalized cuts to cluster shots into scenes. Zhai and Shah [9] presented a framework based on Markov Chain Monte Carlo (MCMC) to detect scenes over home videos and feature films. Sundaram *et al.* [33] proposed to use multimodal features to segment scenes in movies. They first detected audio scenes and video scenes separately. Subsequently, the correspondences of audio and video scenes are established by using a time-constrained nearest-neighbor algorithm. Recently, research efforts attempted to exploit production knowledge to segment scenes or structure programs. Adams *et al.* [34] proposed a so-called "tempo" to segment movies. Tempo is computed at the shot level based on the shot length and motion content. Duan *et al.* [27] incorporated the production rules to segment commercial breaks, in which mid-level features such as visual frame marked with product information (FMPI) and audio scene change (ASC) were developed to segment individual commercials from the perspective of video production. In essence, our proposed solution is an extension of [27] from the commercial domain to the generic TV broadcast domain.

Compared with the segmentation of scenes or stories, program segmentation is a more well-defined task. Scenes and stories are relevant to the interpretation of semantics so that their delineation varies with context knowledge and user experience. On the contrary, the boundaries of TV programs are definite. As a result, many existing methods for scene segmentation do not excel at segmenting programs from TV streams.

D. Multimodal Fusion

In essence, video representation must be multimodal that may involve visual, auditory, and textual channels. It is, of course, beneficial to fuse multimodal features for semantics modeling in multimedia data. Basically, existing fusion strategies can be categorized into two classes: early fusion and later fusion [35]. Early fusion is meant to concatenate wealth of low level features to model a semantic concept through machine learning. For example, an early fusion model was proposed in [36] to fuse the evidences from multiple modalities. Later fusion often involves global fusion schemes and (non)linear fusion schemes. Global fusion models are comparatively flexible yet at the price of more complexity and heavier computational cost on estimating considerable parameters. In general, the parameter estimation of global fusion models is much more difficult than linear fusion models. In [37], a super-kernel early fusion scheme, relying on principle component analysis (PCA) and independent component analysis (ICA), was proposed to fuse modalities in a nonlinear manner. In terms of later fusion, linear fusion models [38] have been widely applied to combine different uni-model classifiers (more popular in TRECVID [39]), in which those combination weights may be learnt from logistic regression. Compared with early fusion, later fusion schemes often yield better results in high-level concept learning [35], whereas more learning computation are incurred. In our work, we will evaluate the effects of both early fusion and later fusion on the program segmentation.

III. Overall Framework

Program segmentation aims to detect the transitions between TV programs, including the boundaries between spots in commercial breaks. Our proposed framework is designed to address various kinds of TV programs, such as news, weather, commercial, sitcoms, sports, and so on. The basic idea is to model common structure characteristics inherent to broadcast program production, i.e., $\langle$ POIM images, program content, POIM images $\rangle$. An illustrative example is given in Fig. 1. Let us take a look at the key frames of a video segment from the CCTV4 channel shown in Fig. 1. Both news and sitcom programs start with POIM images that visually highlight the titles of TV programs. The POIM images at the ends of a spot clearly highlight what is offered by providing explicit information of advertised products. In addition to such visual patterns, the transitions from news to commercials incur prominent audio scene changes from speech to music, and textual content changes from speech in news to voiceover/music in commercials. Hence, we exploit multimodal features to determine the boundaries of programs.

As illustrated in Fig. 1, our solution combines POIM images, audio scene changes, and textual content changes to identify the boundaries of TV programs. It is assumed that the transitions between programs always occur at the video shot transitions (i.e., cuts, fade-in/-out, dissolves, etc.). In practice, we estimate the global motion to detect shot boundaries [40], and extract the average intensity of motion vectors from B- and P-frames in MPEG videos to measure motion so as to select key frames at the local minima of motion within a shot, for the POIM images always appear in still images to visually alert TV viewers, and accordingly we can remove those keyframes with strong animation or digital effects rich in local motions.

From each shot, we extract audiovisual features and textural features according to ASR transcripts or MT transcripts (For CNN, NBC, and MSNBC channels, we use ASR scripts directly. For NTDTV and CCTV4 channels, we use the MT scripts.) An SVM classifier is trained to identify POIM images from wealth of keyframes. A multiscale K–L measure is used to align audio and visual scene changes, and K–L distance is subsequently employed to measure the audio similarities. In terms of textual content, we employ LSA to recover the potential topics of texts, and the cosine distance is used to measure the text-based similarity between adjacent shots. Finally, the fusion of multimodal features is carried out to determine the true boundaries of TV programs from many candidate points.

To facilitate program-oriented browse and search, we propose a novel representation of TV programs based on meaningful key frames and representative words. In addition to meaningful POIM images, representative images are selected by applying the graph based spectral clustering with temporal constraints to key frames. To make proper use of textual information, we employ the Okapi BM25 relevance scoring formula [41] to rank the words from ASR transcripts or MT transcripts. The top 50 words are empirically selected as the most representative words to describe program content. The combination of visual and textual representations reveals the program content in a cross-modal manner, which enables users to readily search and browse their favorite programs from large-scale TV broadcast archives.

IV. Multimodal Features

A. Program Oriented Informative Images

As introduced above, POIM images are a sort of significant indicator to structure TV programs. They may appear at the shots of opening credits and (or) closing credits as shown in Fig. 2. In general, a POIM image consists of texts, graphics, and storytelling embedded images. Texts may include the program titles and the sponsor names. Graphics or images provide an abstract or symbolic representation of regular program contents or advertised products/services. In a sense, POIM images in opening credits can effect the viewers' intention to watch or ignore a program. In commercial videos POIM images may communicate clear visual messages on advertised products/services.

Based on the program genre, POIM images can be classified into four regular categories: news POIM, commercial POIM, sitcom POIM, and others. Examples are given in Fig. 2. A News POIM usually displays the title of a news program, such as "DECISION 2004" and "ELECTION NIGHT 2004." A commercial POIM contains the brand name, trademark, address, telephone number, and cost, etc. For example, "GEICO" is a brand name and "XEROX" is a sponsor company name. The product image might be placed with computer graphics techniques. A sitcom POIM shows the title of sitcoms and the producer information. Other POIM produces the title of weather report (e.g., "EARTHWATCH") or TV station logos (e.g., "MSNBC" and "LBC"), etc.

The presence or absence of POIM images is relevant to program structure. For regular programs, POIM images often appear in the beginning and ending shots of each individual program video section. In commercial videos, POIM images are often presented in the last one or two shots, or occasionally appear in the beginning shot. For the convenience of description, we define the shot containing at least one POIM image as a POIM shot. A POIM shot is utilized to alert us at the beginning of a new section of programs such as news, comedy, drama, and tragedy. Accordingly, our attentions or memories are associated with a continuous episode until an ending POIM shot followed by commercial breaks sometimes. Therefore, the POIM shots are considered a sort of indicator of true program boundaries amongst large amounts of candidates comprising of shot transitions.

1) Visual Feature Extraction: Though the visual appearances of POIM images differ in different programs, POIM images exhibit some common patterns such as clear background, rich text-induced texture, and somehow uniform layout of visual components. Hence, we propose to train the recognizer of POIM images based on low-level visual features. An SVM based binary classifier is thus employed to distinguish POIM images from non-POIM images.

As the regular layout is a sort of significant visual feature in POIM images, we propose to extract local features by image partitions in addition to global features, as shown in Fig. 3. For global features we take into account color and edge cues. The CIE LUV color space is employed because of its perceptual uniformity. Each color channel is uniformly quantized into 100 bins. Three maximum bin values are selected as dominant color features from L, U, and V channels, respectively. This results in a 9-dimensional global color feature vector. Edge direction histogram is employed to capture the statistics of edge features. Canny edge detector [42] is employed with the parameters of sigma $\sigma = 1$ and Gaussian mask $\text{Size} = 9$. Edges are broadly grouped into h categories according to edge orientation with an angle quantizer as

$$A_i = \left[\left\lfloor \frac{180}{h} \right\rfloor \cdot i, \left\lfloor \frac{180}{h} \right\rfloor \cdot (i+1) \right), \quad i = 0, \ldots, h-1. \tag{1}$$

We empirically set $h = 4$ so that a 4-dimensional edge direction feature vector is yielded. The resulting global feature vector is a 13 dimensional one including color and edge features.

For local features, we divide an image into 16 subimages of the equal size followed by feature extraction involving color, edge, and texture to represent each subimage. Within a subimage, the maximum bin value of the color histogram in each color channel is selected as the local dominant color feature. It is worthy noting that the bin values, irrespective of concrete color values, represent the spatial coherency of color. The edge density feature for each subimage is calculated as

$$\text{Edgedensity}_i = \begin{cases} \frac{2E_i}{N}, & \text{if } \frac{E_i}{N} \leq 0.5 \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

where $i = 1, \ldots, 16$ indicates different subimages. E_i is the number of canny edge pixels for subimage i. N is the total number of pixels in subimage i. The local edge density features are 16 dimensional. As Gabor filters [43] exhibit optimal location properties in the spatial domain as well as in the frequency domain, we utilize them to capture the local texture features. In the experiments, we select four Gabor filters with four different orientations at one scale to represent the spatial homogeneousness for each local region. We take the averaged responses of Gabor filters as the local texture feature, and thus we get a $4 \times 16 = 64$ dimensional local texture features.

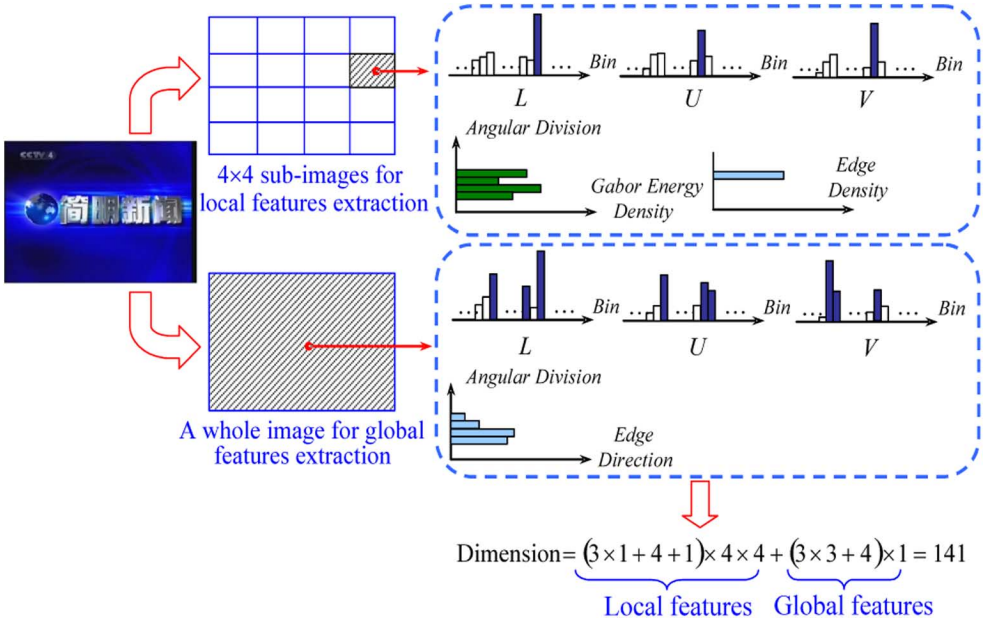


Fig. 3. Illustration of the low-level visual feature extraction for POIM image recognition.

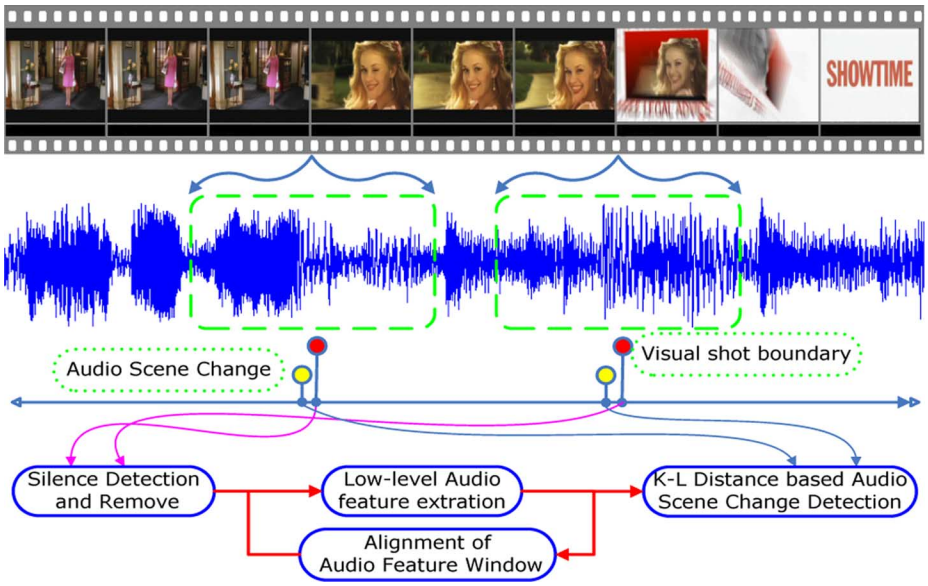


Fig. 4. Procedure for detecting audio scene changes.

2) POIM Image Recognizer: Finally, we obtain in total 141-dimensional low level visual features involving 128-dimensional local features and 13-dimensional global features. C-SVM [24] is subsequently employed to train the POIM classifier based on the low-level visual features. We have compared the classification performance by using different kernels such as linear, polynomial, radial basis function (RBF), and sigmoid. Using the RBF kernel yields the best results in our experiments. To fuse POIM features with other features for subsequent boundary identification, we make use of the probability output of the POIM classifier. In practice, the POIM recognizer is applied to the key frames of each shot due to two advantages: 1) reducing computational cost and 2) avoiding distracting frames from animation effects.

B. Audio Scene Change

The most common TV program is a combination of background music, sound effects, environmental noise, voice-over narration, and storytelling video. Different TV programs often differ in audio characteristics (foreground and/or background). Hence, an appropriate audio scene change model helps to detect program boundaries. "Audio scene" is referred to as a segmented unit by classifying the audio track of a video into pure speech, pure music, speech with music background, silence, etc. [44]. Accordingly, "audio scene change" is associated with the transitions amongst major sound categories or different kinds of sound in the same category (e.g., speaker changes in news videos).

Fig. 4 illustrates the framework to detect and represent audio scene changes. We are interested in determining the occurrences

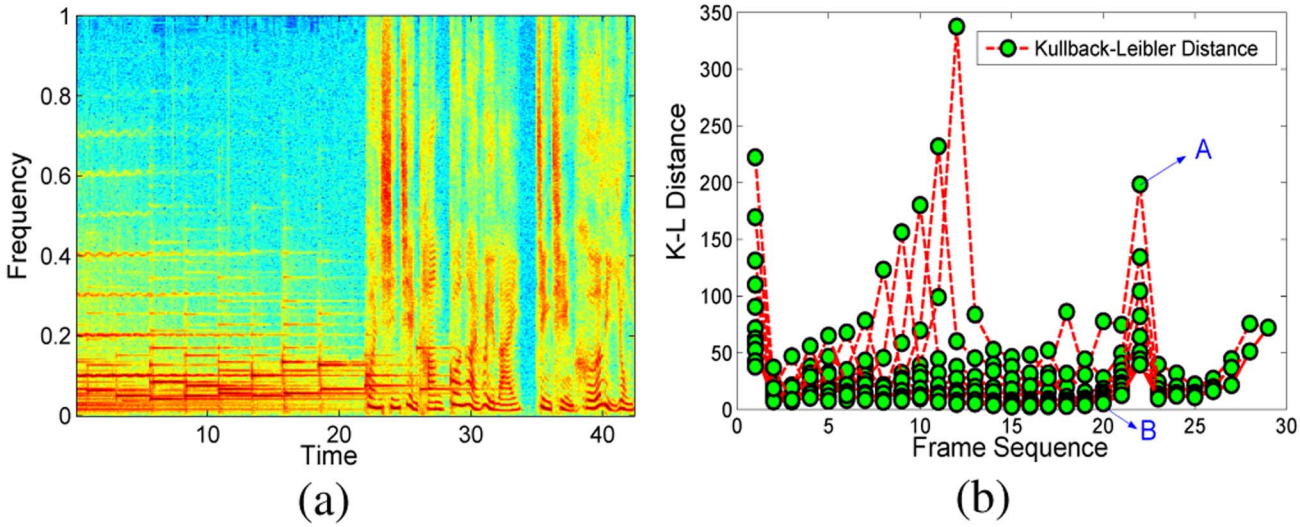


Fig. 5. Illustration of K–L distances between a series of successive audio analysis windows. (a) Spectrogram of an audio clip with the occurrence of true audio scene change. (b) Shifting of the audio scene change point from point "B" to point "A" based on a set of uniform difference peaks.

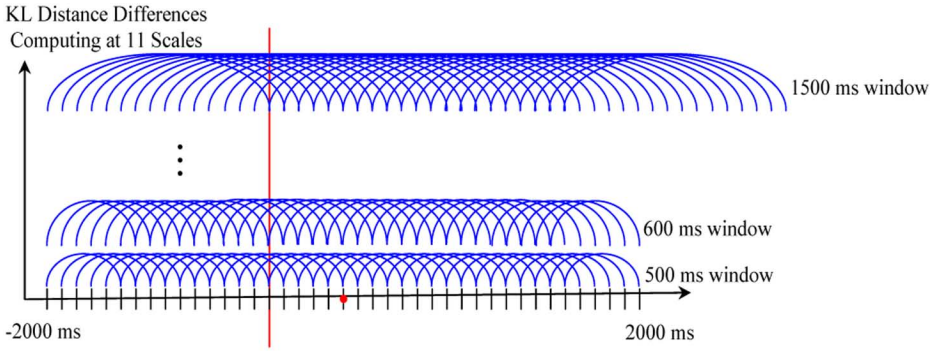


Fig. 6. Illustration of audio analysis windows for computing multiscale K–L distances.

of audio scene changes at candidate program transitions. The size of temporally sliding window is critical for proper ASC detection. Small window size results in unreliable Gaussian-based p.d.f. estimation due to deficient data, while large windows size might lead to more missed detection. Our solution is to apply a multiscale window scheme. Undoubtedly, it is difficult to accurately model and identify each kind of audio scene changes through model-based learning due to exhaustive samples required and subjectivity of labeling data. Hence, we resort to a simple metric-based approach. A multiscale K–L based audiovisual alignment is completed to seek the most probable points of audio scene changes [40]. Optionally, HMM-based models can be trained to characterize the presence or absence of audio scene changes, regardless of the detailed type of audio scene transitions.

1) Silence Removal: Silence segments are present in regular programs, especially at the program transitions. The length of silence segments is variable. The similarity computing between neighboring audio segments has assumed that each segment is a homogeneous one. So the presence of silence segments may generate more false alarms. In experiments, we consider the silence segments lasting for at least 0.3 s and apply a background Harming energy threshold to remove silence segments from original audio signals.

2) Audio Features: Our ASC detection approach considers 43-dimensional audio features comprising Mel-frequency cepstral coefficients (MFCCs) and its first and second derivates (36 features), mean and variance of short time energy log measure (STE) (2 features), mean and variance of short-time zero-crossing rate (ZCR) (2 features), short-time fundamental frequency (or Pitch) (1 feature), mean of the spectrum flux (SF) (1 feature), and harmonic degree (HD) (1 feature) [44]. Due to the dynamic nature of complex sounds, we divide the audio signal into many successive 20 ms analysis frames obtained by shifting a 20 ms sliding window with an interval of 10 ms. Within each 20 ms analysis frame, we compute the features of STE, ZCR, SF, and Harmonic peaks once every 50 samples at an input sampling rate of 22 050 samples/s wherein the sliding window duration is set to 100 samples. Means and variances of STE and ZCR are calculated for seven results from seven overlapped frames while mean of SF is calculated for six results from seven neighbor frames. HD is the ratio of the number of frames that have harmonic peaks to the total number of seven. Pitch and MFCCs are computed directly from each 20 ms frame.

3) Audio Scene Change Characterization With Audiovisual Alignment: The alignment seeks to locate the most probable position of audio scene change within the neighborhood of a transition. As shown in Fig. 6, a multiscale K–L distance metric

[25] is used to evaluate the changes between successive analysis windows. Fig. 5 shows the different locations of change peaks in the cases of different window sizes. Since it is impossible to get *a priori* of window size, we propose a multiscale strategy. That is, we first use different window sizes to yield a set of difference value sequences; each sequence is then scaled to [1] through dividing values by the maximum of each sequence. The most likely audio scene change is located by seeking the highest accumulated difference values derived from these sequences. As illustrated in Fig. 5, a set of uniform peaks associated with a true audio scene change are finally located with about 240 ms delay. The shift of adjusted change point is empirically confined to $[-500\ \mathrm{ms}, 500\ \mathrm{ms}]$.

After candidate ASC points are repositioned by a multiscale audiovisual alignment process, two consecutive audio segments of 2 s before or after the adjusted change point are selected. The K–L distance between these two audio segments is computed to make a final decision on ASC by simply thresholding.

C. Textual Content Similarity

The occurrences of both audio and visual changes cannot guarantee the happening of program transitions, especially in sitcoms or movies. A program comprises the complex stories and each story consists of different scenes. However, those seemly different scenes are often relevant in the context of speech contents, namely, those ASR or MT transcripts. In particular, ASR transcripts have been successfully applied to news video story segmentation [45]. Therefore, we attempt to exploit textual information to address the deficiency of audiovisual features.

We employ latent semantic analysis to model textual information. LSA is a powerful technique in natural language processing to extract semantics [26]. LSA uses a term-document matrix to describe the term occurrences in documents, and it is assumed that there is some underlying or latent structure (i.e., latent semantics) in word usage, which is partially obscured by variability in word choice. The structure in word usage across documents can be obtained by truncated singular value decomposition (SVD) [46].

As different programs vary in the text topic, the inter-program and intra-program similarities of text contents could be different. Textual information is consequently exploited to complement the visual and audio channels aiming to represent the shot-level similarity within a program and the shot-level dissimilarity between different programs.

We recover texts from ASR or MT transcripts for each shot. As a TV program usually lasts for at least five shots, we set the granularity (i.e., a meaningful text content unit) of five shots to represent textual contents. To measure the textual content similarity between adjacent shots, we extend a shot's text content units by expanding two shots forward and backward, respectively. This expansion is to compensate deficient words of each shot and to alleviate the negative effects from the misalignment of audiovisual data. We regard each text content unit as an individual document, and project it to a latent semantic space. Stopping word removal and Porter stemming are performed before producing a vector representation of words in these documents.

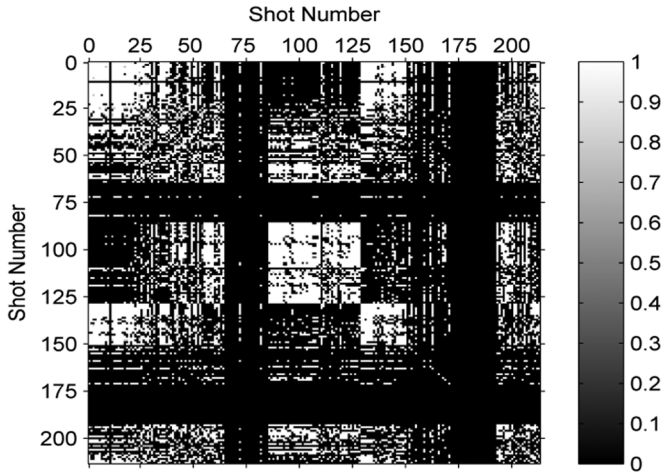


Fig. 7. Example of the normalized textual similarity matrix of a video sequence.

The Okapi BM25 relevance scoring formula [47] is employed to generate a text-vector as

$$\mathrm{weight}_t = tf_s \times \frac{\log\left(\frac{N-n+0.5}{n+0.5}\right)}{k_1 \times \left((1-b)+b\times\frac{dl}{avdl}\right)+tf_s} \quad (3)$$

where tf_s is the Term Frequency occurring in a shot document, the constant factors k_1 and b are set as 2.0 and 0.75, respectively. N is the total number of shot documents, and n is the number of shot documents containing at least one occurrence of t, dl is the length of a shot document and $avdl$ is the average length of shot documents. The document-by-terms matrix $A_{t\times s}$ for a whole video is constructed by shots $\times$ words. The SVD projection is consequently computed by decomposing the matrix $A_{t\times s}$ into the product of three matrices

$$A_{t\times s} = T_{t\times n} S_{n\times n} (D_{s\times n})^T \quad (4)$$

where t is the number of terms, s is the number of shot documents, $n = \min(t, s)$, T and D are orthonormal, i.e., $T^T T = D^T D = I$. Only the first $k(k < n)$ eigenvalues are used to form the pseudo-document matrix

$$\hat{A}_{t\times k} = T_{t\times k} S_{k\times k} (D_{s\times k})^T. \quad (5)$$

In our implementation, $k = 100$ is applied to reconstruct the original shot document matrix. Each column in the pseudo-document matrix is the text vector of a shot document. Cosine distance is applied to measure the textual content similarity at the shot-level. The shot-to-shot textual similarity is

$$\begin{aligned}\hat{A}^T_{t\times k} \times \hat{A}_{t\times k} &= D_{s\times k} S^2_{k\times k} (D_{s\times k})^T \\ &= D_{s\times k} S_{k\times k} (D_{s\times k} S_{k\times k})^T. \end{aligned} \quad (6)$$

Fig. 7 gives an example of similarity matrix from the shot documents of 0.5 h video that comprises two newscasts, 11 commercials, one weather report, and one sitcom. Usually, the beginning shot documents of both news and sitcoms are more similar to the shot documents of commercials because of deficient words from ASR transcripts. It also implies that textual content

similarity alone cannot accurately locate the boundaries of programs. We have noted that most boundaries of spots are missing as few words can be recovered by ASR techniques. Thus, it is significant to incorporate audiovisual features. As the textual content similarity is effective for representing the content continuity of regular programs, it is useful to detect commercial breaks in TV streams as indicated in Fig. 7. As a result, with textual information, the shot-level similarity within a program and the shot-level dissimilarity between different regular programs are enhanced.

V. Fusing Multimodal Features to Segment TV Programs

Both early fusion and later fusion are evaluated towards TV program segmentation. For early fusion, we concatenate the low level visual, audio and textual features to form a complete feature vector, and train an SVM-based binary classifier to recognize program boundaries. For the shots n and $n+1$, their visual features are 141 dimensional respectively. In total a visual feature of 282 dimensions is thus formed. A similar feature concatenation manner applies to audio and textual features. Finally, we obtain a feature vector of 568 dimensions. There have been a variety of supervised learning approaches to learn the association between high-level semantics and low-level multimodal features. In the context of TV program segmentation, a desirable learning algorithm should be able to deal with a few examples, and should be robust across different channels, and be adaptive to unknown types of programs, and should tolerate erroneous recovered data. In our work, SVM is proven to be a solid approach, in which the RBF kernel and linear kernel are used. In addition, to obtain the optimal SVM parameters, we perform the grid-based search within predetermined ranges.

For later fusion, both linear fusion and SVM-based fusion are evaluated and comparison analysis is given. Linear fusion models have been widely used to fuse multimodal information because of its simplicity and good performance. Therefore, we exploit a linear fusion model to integrate the visual, audio, and textual information. As introduced above, the boundaries of TV programs occur at those time-stamp points that produce high occurrence probability of POIM shots, audio scene changes, and textual content changes. On the other hand, we can regard program boundaries as those points that yield lower similarities between adjacent shots, so that we may compute the similarities between adjacent shots based on POIM, audio scenes, and textual contents respectively, to measure the possibility of program boundaries. Three resulting similarities are normalized to the range of [1]. We finally use a weighted linear model to fuse them as follows:

$$S_{n,n+1}\left(\{x_i\}_{i=1}^{k};\{\omega_i\}_{i=1}^{k}\right)=\sum_{i=1}^{k}\omega_i s_{n,n+1}(x_i) \qquad (7)$$

where $k=3$, x_i is the uni-modal feature vector, ω_i is the weight of each uni-model. $s_{n,n+1}$ is the uni-model similarity between the n and $n+1$ shots. $S_{n,n+1}$ is the integrated similarity between the n and $n+1$ shots.

Audio similarity is measured by the ASC output, which is the reciprocal of K–L distance between two neighboring 2-s windows. Textual content similarity is calculated by the cosine distance between the shot document vectors. For POIM-based similarity, we convert the output of the SVM based POIM recognizer to a probability representation [24], and the similarity between the n and $n+1$ shot is defined as: $1/\max(p_n, p_{n+1})$, where p_n and p_{n+1} are the probabilities of relevant key frames being POIM images. If any shot generates a high probability, this point will be declared to be a boundary candidate.

For the linear fusion, weights $\omega_i, i=1,2,3$ are used to coordinate the roles of visual, audio, and text features, and they have close effect on the final performance. According to our experiments, equal weights are unreasonable as POIM images have a higher impact on the result than audio and text features. But audio scene change and textual content similarity can help to reduce those false alarms from erroneous POIM classification. In the experiments, we have $\omega_1=0.44$, $\omega_2=0.37$, $\omega_3=0.19$. An optimal threshold 0.57 is empirically applied to decide the true boundaries finally.

Also we have tried to employ an SVM-based approach to fuse those similarities. We concatenate the similarity outputs of the mid-level visual, audio, and textual feature extractors to form a complete feature vector. For the n and $n+1$ shots, the feature vector comprises the visual POIM features p_n and p_{n+1}, the audio scene similarity $S^a_{(n,n+1)}$ and the textual content similarity $S^T_{(n,n+1)}$. Likewise, the RBF and linear kernel is employed, and the optimal SVM parameters are determined by the grid-based parameter search.

VI. Multimodal Representation of TV Programs

Based on the program segmentation, we propose a multimodal representation of TV programs by determining key frames and representative words. The multimodal representation enables users to readily access and browse interesting programs from large video corpus that aims to maintain desirable content coverage while reducing redundancy. The use of POIM images has differentiated our visual representation from the traditional story board for browsing programs. Besides POIM images, some representative frames are further selected from key frames by the spectral clustering [48]. Moreover, we rank the term weights to select the representative text words from ASR or MT transcripts.

The selection of representative images involves the features of shot similarity and shot duration. Spectral clustering is used to exploit the features and select representative images. The shot similarity takes into account color and temporal similarities between the sets of key frames. Color similarity is computed by histogram matching. The so-called temporal similarity assumes that two shots farther apart in the temporal dimension are less likely to belong to the same cluster. A weighted graph $G=(V,E)$ is constructed for the shots of a given program. L1 distance is applied to define the weights in the graph, as follows:

$$\begin{aligned} W(i,j) &= W_c(i,j)\times W_t(i,j) \\ &= \prod_{k=1}^{36}\exp\left(-\frac{|H_{ik}-H_{jk}|}{\sigma_c}\right)\exp\left(-\frac{|f_i-f_j|}{T\sigma_t}\right) \end{aligned} \qquad (8)$$

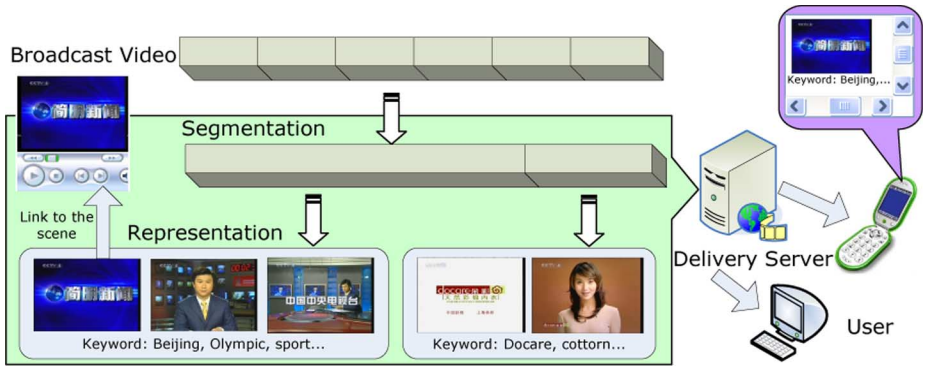


Fig. 8. System configuration of a broadcast video processing prototype.

where f_i denotes the number of keyframes in shot i. H_{ik} and H_{ik} are the kth bins of histograms H_i and H_j, respectively, where we construct a normalized color histogram with 36 bins in HSV space. T is the program duration. We empirically set $\sigma_c = 0.5$ and $\sigma_t = 0.4$. The image selection algorithm is described as below:

Algorithm 1 Selecting representative images

1) Form the affinity matrix $A \in R^{n\times n}$, $A_{i,i} = 0$, $A_{i,j} = W_{i,j}$ (i,j), $D_{i,i} = \sum_j A_{i,j}$, and $L = D^{-1/2}AD^{-1/2}$.

2) Form the matrix $X = [x_1 x_2 \ldots x_k]$ by staking the k largest eigenvectors of L.

3) Normalize each row of X to unit length.

4) Cluster each row X into k clusters via dynamic K-mean.

5) Assign to each node i the cluster number corresponding to its row.

6) Rank keyframes by shot duration for each cluster.

7) Add the POIM images as representative images.

Currently, one key frame is selected from each cluster to the set of representative images. We apply a simple ranking criterion. Firstly, a longer shot should be given more importance weights in video scenes. Secondly, representative words can be determined by ranking the term weights in (3). We empirically select the top 50 words.

As illustrated in Fig. 8, we have developed a prototype system to process TV streams, which involves the segmentation, representation and delivery of TV programs. The resulting representative frames are associated with the relevant video scenes in a Hyper-link, which enables users to readily browse and navigate programs, and efficiently seek the favorite scenes over various devices such as mobile phones or personal computers. The prototype is implemented by Windows Media SDK.

VII. Experiments

Our experiments were conducted over TRECVID 2005 video corpus, which is an open benchmarking dataset for feature extraction and video search evaluation. The training and testing dataset are collected from five TV channels: CNN, NBC, MSNBC, NTDTV, and CCTV4. Those programs have covered various genres including news, commercial, movie, sitcom, animation, MTV, weather report, and so on.

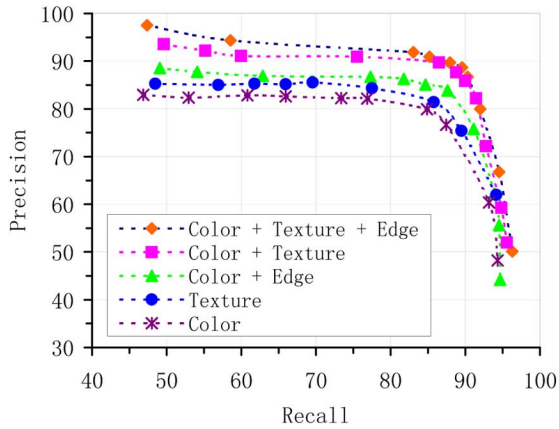


Fig. 9. POIM image classification results by using different visual features and different SVM kernel parameter pairs of (γ, c).

A. Results of Poim Image Classification

In order to evaluate the POIM image recognizer, we collect 6000 frames including 2000 POIM images and 4000 non-POIM images. Some 1000 POIM images and 2000 non-POIM images are randomly selected as the training set, and the rest as the testing set. The ground truth of POIM images is manually established. F1 measure is used to evaluate the performance

$$\mathrm{F1} = \frac{2 \times \mathrm{Recall} \times \mathrm{Precision}}{\mathrm{Recall} + \mathrm{Precision}}. \tag{9}$$

A RBF kernel, $\exp\left(-\gamma \|x_i - x_j\|^2\right) \gamma > 0$, is used for SVM learning. There are four parameters, i.e., gamma γ, cost c, class weight ω_i, and tolerance e. Class weight ω_i is to deal with the problem of data unbalance so that the cost c of class i is set to $\omega_i \times c$. We set $\omega_1 = 1$ and $\omega_1 = 2$ for POIM and non-POIM, respectively. Tolerance e is set to 0.0001. Gamma γ is tuned between 0.1 and 10 and cost c is tuned between 0.1 and 1. Fig. 9 shows the POIM classification performance from averaging the results of ten trials. The curves of "Color" and "Texture" have demonstrated the individual capabilities of color and texture features in distinguishing POIM from non-POIM. Comparatively, texture features play a more important role. The combination

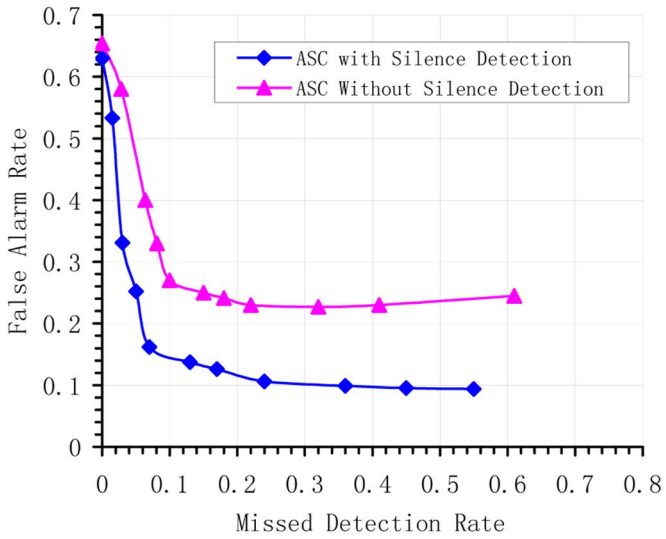


Fig. 10. Two DET curves of ASC detection with and without removing silence segments.

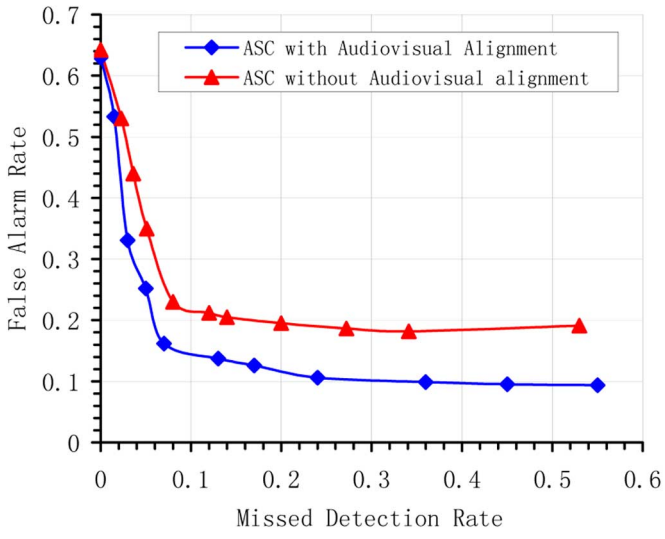


Fig. 11. Two DET curves of ASC detection with and without the process of audiovisual alignment.

of color, edge, and texture features results in a significant improvement of performance, and it has achieved an accuracy of $\mathrm{F1} = 90.2\%$. Towards TV program segmentation, the recall rate of our POIM recognizer is important as a higher recall can recover more candidates to be further verified by fusing multimodal features. As shown in Fig. 9, our POIM classifier has achieved a promising recall up to 95% with a pair of optimal parameters $(\gamma, c) = (0.5, 1.0)$.

B. Results of Detecting Audio Scene Change

To evaluate the performance of audio scene change detection, we employ the detection error tradeoff (DET) curve based on false alarm rate (FAR) and missed detection rate (MDR), as follows:

$$\mathrm{FAR} = \frac{\text{Number of false alarm ASC}}{\text{Number of detected ASC}} \tag{10}$$

and

$$\mathrm{MDR} = \frac{\text{Number of missed ASC}}{\text{Number of target ASC}}. \tag{11}$$

We first investigate the effects of silence segment removal on ASC detection. The results are listed in Fig. 10. The minimal length of valid silence segments is chosen to be 0.3 s in detecting silence. With the increasing threshold for deciding silence, MDR increases while FAR decreases. Given a fixed MDR $(\mathrm{MDR} > 0.1)$, FAR is improved by 10% when the removal of silence segments is applied and considerable false alarms are reduced. Fig. 11 shows the effects of the audiovisual alignment process on ASC detection. Without the alignment, we extract audio features from a 4 s window exactly centered at a shot boundary followed by K–L distance computing between two consecutive 2 s windows to detect ASC. With the alignment, given a fixed MDR $(\mathrm{MDR} > 0.08)$, FAR is improved by some 7%. Here, both silence removal and audiovisual alignment significantly contribute to a robust ASC detection as lots of false alarms are rejected.

C. Results of TV Program Segmentation

In this section, we evaluate the performance of program segmentation. A program is often interrupted by commercial breaks in our empirical dataset. The ground truth of program boundaries is manually established. Both precision and recall are applied to evaluate the performance. The recall rate indicates the capability of recovering true program boundaries, while the precision rate indicates the capability of eliminating false detections. For the convenience of feature extraction, we concatenate the piecewise video segments collected from the same channel to form a long video sequence. Finally we generate a video segment about 15 h for each channel (i.e., CNN, NBC, MSNBC, NTDTV, and CCTV4).

First, let us evaluate the impact of multimodal features on the performance. Table I lists the results via linear fusion through tuning parameters over the whole data set of each channel. Note that cross-validation is applied in Table II to compare different fusion strategies. From Table I, we can observe that the textual content similarity alone is unable to accurately locate program boundaries, and most errors derive from the presence of spots. However, textual similarity is a useful utility when both audio and visual features failed to identify true boundaries. POIM is a very useful visual feature, which is shown by the promising performance of $\mathrm{F1} = 0.73$ that greatly outperforms the result of $\mathrm{F1} = 0.57$ with textual features only.

Table II lists the results of three-fold cross-validation for each fusion strategy. The average performance of three trials is calculated. Different feature combinations have been tried out such as "POIM" + "ASC" and "POIM" + "ASC" + "TCS." Overall, both precision and recall have been improved by combining features. SVM has been employed in both early fusion and later fusion. In early fusion, the RBF kernel obtains better performance than linear kernel with 568 dimensional low level features. Compared with early fusion, later fusion yields better results. For later fusion, the SVM-based fusion with a linear kernel has achieved better results than a radial basis function (RBF)

TABLE I
PERFORMANCE OF TV PROGRAM SEGMENTATION IN THE CASES OF DIFFERENT MULTIMODAL FEATURE COMBINATIONS. (TCS: TEXTUAL CONTENT SIMILARITY, ASC: AUDIO SCENE CHANGE, VP: VISUAL POIM, P: PRECISION, R: RECALL)

	MSNBC			NBC			CNN			NTDTV			CCTV-4		
Features	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
TCS	0.54	0.53	0.53	0.58	0.54	0.56	0.62	0.53	0.57	0.61	0.54	0.57	0.62	0.58	0.60
VP	0.68	0.76	0.71	0.79	0.73	0.75	0.72	0.76	0.73	0.71	0.77	0.73	0.67	0.78	0.72
VP+ASC	0.86	0.90	0.88	0.84	0.87	0.85	0.85	0.91	0.87	0.84	0.92	0.88	0.85	0.91	0.87
VP+ASC+TCS	0.90	0.92	0.91	0.89	0.90	0.89	0.88	0.91	0.89	0.86	0.93	0.89	0.87	0.90	0.88

TABLE II
PERFORMANCE OF TV PROGRAM SEGMENTATION IN THE CASES OF BOTH EARLY FUSION AND LATER FUSION SCHEMES

		Precision	Recall	F1
Early fusion (RBF kernel)		0.75	0.69	0.72
Early fusion (Linear kernel)		0.68	0.71	0.69
Later fusion	SVM fusion (RBF kernel)	0.87	0.88	0.87
	SVM fusion (Linear kernel)	0.90	0.93	0.91
	Linear fusion	0.89	0.93	0.91

TABLE III
PERFORMANCE OF TV PROGRAM SEGMENTATION OVER FOUR INDIVIDUAL TESTING CHANNELS

TV Channels	Precision	Recall	F1
CCTV-1	0.92	0.93	0.92
Hunan	0.89	0.92	0.90
CCTV-3	0.84	0.87	0.85
CQTV	0.87	0.94	0.90

kernel. The simple weighting-based linear fusion scheme have achieved a result of $\mathrm{F1} = 0.91$ comparable to the SVM-based fusion with a linear kernel, which shows that the simple linear classifier with empirical parameter tuning is able to achieve the promising results comparable to that by using advanced machine learning algorithms, thanks to the effective feature reduction by developing mid-level features.

Undoubtedly, a desirable approach to program segmentation should be robust across different TV channels. As there are too many TV channels and each channel have fairly diverse programs, the extensibility of our proposed algorithm is significant for its usability and applicability. As listed in Table I, a comparable performance has been achieved over all the channels. The reason is that TV programs share some common patterns in terms of temporal and structural characteristics regardless of different TV channels. Although our experiments were conducted over limited TV channels, we argue that the proposed algorithm is robust across more different channels, because TV broadcast video production almost always follows some basic production rules irrespective of either TV channels or program content. To provide some empirical results, we conduct the following experiment. With the parameters and thresholds tuned for the above mentioned five TV channels, we test the developed boundary recognizer over unknown video data collected from several new channels such as CCTV-1, Hunan, CCTV-3, and CQTV. Each video sequence has a length of 2 h. As listed in Table III, the average performance $\mathrm{F1} = 0.89$ is obtained by applying the original experimental settings. Compared with the results over other TV channels, the performance of CCTV-3 channel is worst. The errors are mostly caused by the introduction clips of upcoming programs or events, which yield considerable missed detections. Such introduction clips are always longer than commercial videos and behave more like a mini story whereas less uniform format exhibits.

For comparison purposes, we list some results of program segmentation with the hierarchical approach [15] in Table IV. The hierarchical approach in [15] works at the scene level while our method works at the program level. As discussed above, the definition of scenes is somehow subjective while the definition of programs is clear. From Table IV, we can see that our proposed approach has achieved a precision of 0.89 and a recall of 0.93 on average, while for the hierarchical approach, only a precision of 0.81 and a recall of 0.71 result. Some false alarms come from the program lead-in/-out logos that may be present for several times during news programs, especially in the presence of quite different stories before or after the program logo shots. Other errors come from those missed program boundaries including commercial boundaries. When a short commercial (say 10 s, only a few shots) is broadcasted successively for several times, persistent POIM shots, similar audio and text make it hard to segment each individual one with our approach. Nevertheless, it is worthy noting that our proposed approach is able to distinguish individual spots in general, which was not systematically addressed in previous approaches.

D. Some Examples of TV Program Representation

Fig. 12 gives some illustrative examples of TV program representation. Fig. 12(a) is on the representation of a news program. The first, the last, and the last second keyframes are POIM images. The first and the last second keyframes have a superimposed title "ELECTION NIGHT 2004" and the last keyframe is marked with the sponsor logo "USTA." The keywords such

TABLE IV
PERFORMANCE OF TV PROGRAM SEGMENTATION WITH TWO DIFFERENT APPROACHES. (C: CORRECT DETECTED BOUNDARIES OF TV PROGRAMS, M: MISSED BOUNDARIES OF TV PROGRAMS, F: FALSE BOUNDARIES OF TV PROGRAMS, P: PRECISION, R: RECALL. WE ONLY LIST FIVE SEGMENTS FOR EACH CHANNEL)

	Programs	News	Commercial	Others	Our approach					Hierarchical approach [15]				
					C	M	F	P	R	C	M	F	P	R
MSNBC_1	25	3	22	0	20	3	2	0.80	0.91	21	4	10	0.84	0.68
MSNBC_2	20	3	16	1	19	1	3	0.95	0.86	16	4	5	0.80	0.76
MSNBC_3	21	3	17	1	21	0	1	1.00	0.95	12	9	4	0.57	0.75
MSNBC_4	14	3	11	0	10	4	1	0.71	0.91	14	0	6	1.00	0.70
MSNBC_5	23	3	19	1	21	2	2	0.91	0.91	20	3	8	0.87	0.71
NBC_1	17	3	12	2	14	3	3	0.82	0.82	13	4	5	0.76	0.72
NBC_2	25	4	21	0	23	2	1	0.92	0.96	21	4	4	0.84	0.84
NBC_3	18	3	14	1	16	2	0	0.89	1.00	15	3	5	0.83	0.75
NBC_4	20	3	16	1	18	2	5	0.90	0.78	15	5	4	0.75	0.79
NBC_5	26	2	24	0	24	2	2	0.92	0.92	22	4	8	0.85	0.73
CNN_1	11	3	8	0	11	0	0	1.00	1.00	8	3	2	0.73	0.80
CNN_2	24	3	21	0	23	1	1	0.96	0.96	16	8	3	0.67	0.84
CNN_3	23	3	20	0	19	4	0	0.83	1.00	15	8	3	0.65	0.83
CNN_4	36	5	31	0	31	5	2	0.86	0.94	28	8	13	0.78	0.68
CNN_5	25	4	21	0	22	3	1	0.88	0.96	21	4	7	0.84	0.75
NTDTV_1	7	4	3	0	7	0	0	1.00	1.00	6	1	4	0.86	0.60
NTDTV_2	13	4	8	1	12	1	0	0.92	1.00	10	3	5	0.77	0.67
NTDTV_3	9	4	4	1	8	1	1	0.89	0.89	6	3	4	0.67	0.60
NTDTV_4	10	4	5	1	10	0	0	1.00	1.00	7	3	5	0.70	0.58
NTDTV_5	12	4	7	1	11	1	0	0.92	1.00	10	2	4	0.83	0.71
CCTV4_1	12	2	8	2	12	0	2	1.00	0.86	8	4	3	0.67	0.73
CCTV4_2	6	1	3	2	6	0	0	1.00	1.00	5	1	3	0.83	0.63
CCTV4_3	8	1	5	2	5	3	0	0.62	1.00	6	2	7	0.75	0.46
CCTV4_4	19	2	15	2	16	3	2	0.84	0.89	16	3	9	0.84	0.64
CCTV4_5	8	1	5	2	6	2	1	0.75	0.86	7	1	3	0.88	0.70
Average								0.89	0.93				0.81	0.71

as "VOTING," "ELECTION," and "CANDIDATE" provide informative texts to complement the visual information of POIM images. Consequently, combining visual and textual information provides an integrated expression for the users to figure out program content. Fig. 12(b) is about the representation of a sitcom "OU RAN." Out of the representative images, we may oversee the main actors and some relevant scenes. The MT transcripts are used to extract text words. The representative words such as "LOVE," "WORK," and "BUSINESS," can tell that it is a sort of affectionate sitcom. Moreover, an example of commercial videos is shown in Fig. 12(c). Such keywords as "PAIN," "DOCTORS," "PILLS," "OSTEOARTHRITIS," etc., are relevant to the purposes of advertised product "CELEBREX." Two POIM images visually present product information. In the cases of few words recovered from the ASR/MT transcripts, we can list all the keywords. Empirical study has shown that our proposed multimodal representation is advantageous to the access, browse, and search of favorite TV programs, although more user study and subjective evaluation are required.

VIII. CONCLUSION

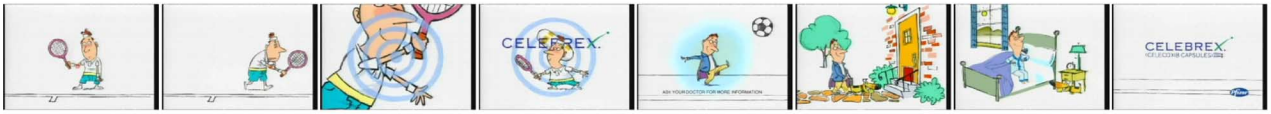
We have proposed a multimodal approach to recover the program oriented temporal patterns for structuring and representing TV streams. To segment TV programs more accurately, we have developed a set of useful mid-level features such as POIM, ASC, and TCS from the perspective of program production. Various fusion strategies have been investigated towards properly characterizing the program transition via multiple modalities. Our empirical study has shown that a linear fusion model is simple but excels in integrating multimodal features to structure TV streams. Compared with the traditional scene-based approaches, our proposed solution has yielded better results in segmenting individual programs. Through testing the well-tuned boundary

KEYWORDS: VOTING, NUMBER, PERCENT, STATE, PRESIDENTIAL, ELECTION, CIVIL, BUSH, GONNA, VIEWER, SENATES, IDEA, THOUSAND, CANDIDATE, TOKEN, JOHN, DEMOCRATIC, HUNDRED, BALLOTS, COUNTRY, POLITICAL, COMING, REPUBLICANS, DIFFERENT, PUBLIC, SEAT, RATE, BILLBOARD, JUDGE, ELECTORAL, NORTH, STATEMENT, PARTY, WORK, CONCERN, DEPARTMENT, ISSUE, EDUCATION, POWER, OPTIONAL, SOUTH, GEORGE, AREA, MEMBERS, INTERVIEWS, REPORT, DEPARTMENT, SECRETARY, POPULATION, DISAPPOINT.

(a)

KEYWORDS: LOVE, WORK, BUSINESS, SAID, LIFE, SHANGHAI, MAKE, HEALTH, MEAL, COME, HAPPY, HOPE, USED, GOOD, WANT, MONEY, SPEECH, REASON, PARENT, ACCIDENT, MOTHER, PERSON, CAREFUL, HOME, CONTROL, LOOK, FATHER, STOP, NAME, FUTURE, KNOW, RECEIVED, SUPPORT, FEEL, SCORED, SEEM, ECONOMIC, FRIEND, CALLED, RETURN, DECISION, WATER, SUITABLE, RICE, GRACEFUL, TEST, CONFIDENCE, SUPPORT, SITUATION, FUNCTION.

(b)



KEYWORDS: PAIN, DOCTORS, ENDANGERED, PILLS, SERIOUS, STIFFNESS, OSTEOARTHRITIS, JOINT, INFLAMMATION, DRUGS, CELEBRATED, VIRTUALLY, STARTED, PRESCRIPTION, SENSITIVE, MAGAZINE, REACTION, SOURCE, PETER, RELIEF.

(c)

Fig. 12. Examples of multimodal representation of TV programs. (a) News Program "ELECTION NIGHT 2004;" (b) Sitcom "OU RAN;" (c) Commercial "CELEBREX." (a) News Program "ELECTION NIGHT 2004;" (b) Sitcom "OU RAN;" and (c) Commercial "CELEBREX."

detector over unknown TV channels, we have shown the extensibility of our solution. In particular, our proposed approach is able to distinguish the boundaries of individual commercials, which provides some basic structural information to digest TV commercials [27]. With the proper segmentation of TV programs, we have come up with a sort of representation to describe program contents by combining visual and textual information. This representation is expected to facilitate the manipulation and management of TV programs.

REFERENCES

[1] J. Masthoff and R. Lukin, "Preface: Workshop future TV: Adaptive instruction in your living room," in *Proc. Future TV: Adaptive Instruction In Your Living Room*, 2002.

[2] Tivo [Online]. Available: http://www.tivo.com

[3] L. Ardissono, A. Kobsa, and M. M. editors, *Personalized Digital Television*. Norwell, MA: Kluwer, 2004.

[4] M. M. Yeung and B. L. Yeo, "Time-constrained clustering for segmentation of video into story units," in *Proc. ICPR'96*, 1996, vol. 3, pp. 375–380.

[5] C. W. Ngo, T. C. Pong, and H. J. Zhang, "On clustering and retrieval of video shots through temporal slices analysis," *IEEE Trans. Multimedia*, vol. 4, pp. 446–458, Dec. 2002.

[6] H. Okamoto, Y. Yasugi, N. Babaguchi, and T. Kitahashi, "Video clustering using spatio-temporal image with fixed length," in *Proc. ICME'02*, Aug. 2002, vol. 1, pp. 53–56.

[7] D. Q. Zhang, C. Y. Lin, S. F. Chang, and J. R. Smith, "Semantic video clustering across sources using bipartite spectral clustering," in *Proc. ICME '04*, Jun. 2004, pp. 117–120.

[8] Z. Rasheed and M. Shah, "Detection and representation of scene in videos," *IEEE Trans. Multimedia*, vol. 7, pp. 1097–1105, Dec. 2005.

[9] Y. Zhai and M. Shah, "A general framework for temporal video scene segmentation," in *Proc. ICCV'05* ., Beijing, China, Oct. 2005, pp. 15–21.

[10] J. Huang, Z. Liu, and Y. Wang, "Joint scene classification and segmentation based on hidden Markov model," *IEEE Trans. Multimedia*, vol. 7, pp. 538–550, Jun. 2005.

[11] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 194–218.

[12] J. Gauvain, L. Lamel, and G. Adda, "The limsi broadcast news transcription system," *Speech Commun.*, pp. 89–108, 2002.

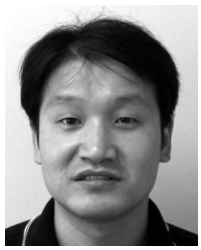
[13] S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," in *Proc. ACM MM'95*, 1995.
[14] A. G. Hauptmann and M. J. Witbrock, "Story segmentation and detection of commercials in broadcast news video," in *Proc. Advances in Digital Libraries Conf.*, Santa Barbara, CA, Apr. 1998.
[15] J. H. Z. Liu and W. Yao, "Integration of audio and visual information for content-based video segmentation," in *Proc. ICIP 98*, Oct. 1998, vol. 3, pp. 526–529.
[16] M. Roach, J. Mason, and M. Pawlewski, "Video genre classification using dynamics," in *Proc. ICCASP 01*, Salt Lake City, UT, May 2001, vol. 3, pp. 7–11.
[17] R. Jasinschi and J. Louie, "Automatic TV program genre classification based on audio patterns," in *Proc. Euromicro Conf.*, Sep. 2001, pp. 370–375.
[18] B. T. Truong and C. Dorai, "Automatic genre identification for content-based video categorization," in *Proc. ICPR'00*, 2000, vol. 4, pp. 230–233.
[19] G. W. L. Agnihotri and N. Dimitrova, "TV program classification based on face and text processing," in *Proc. ICME'00*, 2000, vol. 3, pp. 1345–1348.
[20] T.-H. Kim, W.-H. Lee, and D.-S. Jeong, "Automatic video genre identification method in mpeg compressed domain," in *Proc. ITC-CSCC'02*, Jul. 2002, pp. 1527–1530.
[21] J. Wang, L. Duan, H. Lu, and J. S. Jin, "A semantic image category for structuring TV broadcast video streams," in *Proc. PCM'06 , 2006*, pp. 279–286.
[22] Opencredit [Online]. Available: http://en.wikipedia.org/wiki/Opening_credits
[23] Closecredit [Online]. Available: http://en.wikipedia.org/wiki/Closing_credits
[24] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[25] J. P. Campbell, Jr., "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1642, Sep. 1997.
[26] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Soc. Inform. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
[27] L.-Y. Duan, J. Wang, Y. Zheng, J. S. Jin, H. Lu, and C. Xu, "Segmentation, categorization, and identification of commercials from TV streams using multimodal analysis," in *Proc. ACM MM'06*, 2006, pp. 202–210.
[28] T.-S. Chua, S.-F. Chang, L. Chaisorn, and W. Hsu, "Story boundary detection in large broadcast news video archives: Techniques, experience and trends," in *Proc. ACM MM'04*, 2004, pp. 656–659.
[29] P. Rennert, "Streamesage unsupervised asr-based topic segmentation," in *Proc. TRECVID Workshop*, 2003.
[30] J. Reynar, "An automatic method of finding topic boundaries," in *Proc. 32nd Annu. Meeting of the Association for Computational Linguistics*, 1994.
[31] W. Hsu, S.-F. Chang, C.-W. Huang, L. Kennedy, C.-Y. Lin, and G. Iyengar, "Discovery and fusion of salient multi-modal features towards news story segmentation," in *Proc. IS&T/SPIE Symp. Electronic Imaging: Science and Technology*, Jan. 2004.
[32] A. Hanjalic, R. Lagendijk, and J. Biemond, "Automatically segmenting movies into logical story units," in *Proc. 3rd Int. Conf. VISUAL '99*, Jun. 1999.
[33] H. Sundaram, "Segmentation, Structure Detection, and Summarization of Multimedia Sequences," Ph.D. dissertation, Columbia Univ., New York, 2002.
[34] B. Adams, C. Dorai, and S. Venkatesh, "Automatic extraction of expressive elements from motion pictures: Tempo," *IEEE Trans. Multimedia*, vol. 4, pp. 472–481, Dec. 2002.
[35] C. G. Snoek, M. Worring, J. C. v. Gemert, J.-M. Geusebroek, and A. W. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proc. ACM MM'06*, 2006.
[36] C. G. Snoek, M. Worring, J.-M. Geusebroek, D. C. Koelma, and F. J. Seinstra, "The mediamill trecvid 2004 semantic video search engine," in *Proc. TRECVID Workshop'04*, 2004.
[37] Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith, "Optimal multimodal fusion for multimedia data analysis," in *Proc. ACM MM'04*, 2004.
[38] C. C. Vogt and G. W. Cottrell, "Fusion via a linear combination of scores," *Inf. Retriev.*, Oct. 1999.
[39] Trecvid [Online]. Available: http://www-nlpir.nist.gov/projects/t01v
[40] J. Wang, L. Duan, H. Lu, J. S. Jin, and C. Xu, "A mid-level scene change representation via audiovisual alignment," in *Proc. ICASSP'06*, May 2006.
[41] D. Hawking, T. Upstill, and N. C. Toward, "Toward better weighting of anchors," in *Proc. ACM SIGIR'04*, Jun. 2004, pp. 25–29.
[42] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 6, pp. 679–698, 1986.
[43] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 837–842, 1996.
[44] L. Lu, H. Jiang, and H. Zhang, "Robust audio classification and segmentation method," in *Proc. ACM MM'01*, 2001, pp. 203–211.
[45] W. Hsu, L. Kennedy, C.-W. Huang, S.-F. Chang, C.-Y. Lin, and G. Iyengar, "News video story segmentation using fusion of multi-level multi-modal features in trecvid 2003," in *Proc. ICASSP'04*, May 2004.
[46] J. E. Gentle, "Singular value factorization," *Numer. Linear Algebra Applicat. Statist.*, 1998.
[47] M. Franz, J. S. McCarley, and S. Roukos, "Segmentation and detection at IBM: Hybrid statistical models and two-tiered clustering broadcast news domain," in *Proc. TDT-3 Workshop'00*, 2000.
[48] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inform. Process. Syst.*, 2001.

Jinqiao Wang received the B.E. degree in 2001 from Hebei University of Technology, China, and the M.S. degree in 2004 from Tianjin University, China. He is currently pursuing the Ph.D. degree at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

His research interests include multimedia information retrieval, video/image content analysis and classification, object detection and classification, machine learning, pattern recognition, and computer vision.

Lingyu Duan received the B.S. degree in applied physics from Dalian University of Technology (DUT), in 1996, the M.S. degree in automation from University of Science & Technology of China (USTC), Hefei, China, in 1999, the M.S. degree in computer science from National University of Singapore (NUS), Singapore, in 2002, and the Ph.D. degree in the School of Design, Communication, and Information Technology, University of Newcastle, Australia, in 2008.

He is a Research Scientist at the Institute for Infocomm Research, Singapore. His current research interests include image/video processing, multimedia, computer vision and pattern recognition, and machine learning.

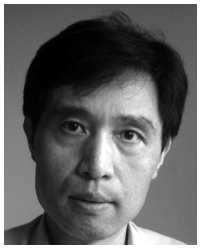


Qingshan Liu (M'06) was born in 1975. He received the M.Sc. Degrees in automatic control from South-East University, China, in 2000. He received the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2003.

He is now an Associate Professor with Chinese Academy of Sciences. From June 2004 to April 2005, he was an Associate Researcher with the Multimedia Laboratory, Chinese University of Hong Kong. He has published more than 70 papers in the journals and conferences. His research focus includes face recognition, facial expression analysis, event-based video analysis, image and video retrieval.

Hanqing Lu (M'06) received the B.E. degree in 1982 and the M.E. degree in 1985 from Harbin Institute of Technology, China, and the Ph.D. degree in 1992 from Huazhong University of Sciences and Technology, China.

Currently, he is a Deputy Director of the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include image and video analysis, medical image processing, and object recognition. He has published more than 100 papers in these fields.



Jesse S. Jin (M'98) was born in Beijing, China, in 1956 and received the B.E. degree from Shanghai Jiao Tong University, China, the M.E. degree from CTU, China, and the Ph.D. degree from the University of Otago, New Zealand in 1982, 1987, and 1992, respectively. All degrees are in computer science.

He held academic positions with the University of Otago, the University of New South Wales, and the University of Sydney, and is the Chair Professor of information technology in the School of Design, Communication, and IT, University of Newcastle, NSW, Australia. He has published 175 articles. He also has one patent and is in the process of filing three more patents. His research interests include multimedia, medical imaging, computer vision, and pattern recognition.

Dr. Jin is a member of ACM, ASTS, and IS&T. He established a spin-off company that won the 1999 ATP Vice-Chancellor New Business Creation Award.