Unsupervised Video-Shot Segmentation and Model-Free Anchorperson Detection for News Video Story Parsing

Xinbo Gao, *Member, IEEE,* and Xiaoou Tang, *Senior Member, IEEE*

***Abstract*—News story parsing is an important and challenging task in a news video library system. In this paper, we address two important components in a news video story parsing system: shot boundary detection and anchorperson detection. First, an unsupervised fuzzy c-means algorithm is used to detect video-shot boundaries in order to segment a news video into video shots. Then, a graph-theoretical cluster analysis algorithm is implemented to classify the video shots into anchorperson shots and news footage shots. Because of its unsupervised nature, the algorithms require little human intervention. The efficacy of the proposed method is extensively tested on more than 5 h of news programs.**

***Index Terms*—Anchorperson detection, fuzzy clustering, graph-theoretical cluster analysis, video library.**

I. Introduction

As DIGITAL video libraries and archives of immense size are becoming accessible over data networks, efficient video retrieval and browsing have become crucially important. The temporally linear and data-intensive nature of digital video necessitates automatic techniques in the library creation and video indexing process. The first step commonly taken for automatic video processing systems is to parse a continuous video sequence into camera shots, which are the basic video units representing continuous action in both time and space in a scene. A number of methods have been proposed to detect the shot boundaries [2], [6], [13]. In general, low-level features such as color histograms [16], [21], [28], [44], motion vectors [1], [7], [33], and compression parameters [25], [31], [41], [47] are used in the parsing process.

After videos are segmented into camera shots, high-level techniques are required to group individual shots into a more descriptive segment of the video sequence and to extract intelligent annotation and indexing information from the segment. However, a universal solution for high-level video analysis is very difficult, if not impossible, to achieve. Researchers have been focusing on specific applications utilizing domain knowledge in videos such as sports programs, movies, commercial advertisements, and news broadcasts. For example, several sports video analysis techniques have been proposed recently. Utilizing prior knowledge of basketball game structure, Tan *et al.* extract high-level annotation for video events such as close-up views, fast breaks and shots at the basket [35]. Gong *et al.* [14] propose a system to parse the content of soccer videos. Using the standard layout of a soccer field they manage to classify a soccer video into various play categories, including "shot at left goal," "top-left corner kick," "in midfield," etc. Chang *et al.* [10] develop an automatic indexing scheme for football video by integrating image and speech analysis techniques. For movie video analysis, a number of high-level movie segmentation approaches have been proposed. Pfeiffer *et al.* [32] extract events such as dialogs and action intensive segments from a movie to make a movie trailer. Yeung *et al.* [42] use time-constrained clustering and predefined models to recognize patterns corresponding to dialogs, action sequences, and arbitrary story units. Based on the assumption that the visual content within a movie episode is temporally consistent, Hanjalic *et al.* [17] segment a movie into logical story units to approximate actual movie episodes. To detect commercials in a video program, Taniguchi *et al.* [36] use a high rate of scene breaks as a distinct characteristic for commercials. Hauptmann *et al.* [20] combine two sources of information, the presence of black frames and the rate of scene changes, to detect commercial breaks in news programs.

Among all the domain-specific applications, news video processing is probably the most extensively studied topic. Broadcast news is valuable to data analysts in the government, information providers, and television consumers [8]. However, since news events happen daily all over the world, a person cannot afford to view all news shows on all channels indiscriminately. To alleviate the problem, a news video database that compresses and digitally stores news broadcasts and provides interactive retrieval interface over a network needs to be developed. This would enable automatic retrieval of relevant news stories from all the networks and news sources covering the topic of interest [19].

In recent years, several news video processing systems have been developed, such as the MEDUSA system [9], the Broadcast News Navigator System [8], [23], [24], [27] and the Informedia Project [19], [20], [29], [37]–[40]. Most of these systems allow automatic or semi-automatic parsing and annotation of news recordings for interactive news navigation, content-based retrieval and news-on-demand applications. For effective news browsing and retrieval, reliable news story parsing is crucial in the video library system. Correct parsing of news stories leads

Manuscript received February 3, 2001; revised April 16, 2002. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region under Grant CUHK4378/99E and Grant AoE/E-01/99. This paper was recommended by Associate Editor F. Pereira.

The authors are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong (e-mail: xgao@ie.cuhk.edu.hk; xtang@ie.cuhk.edu.hk).

Publisher Item Identifier 10.1109/TCSVT.2002.800510.

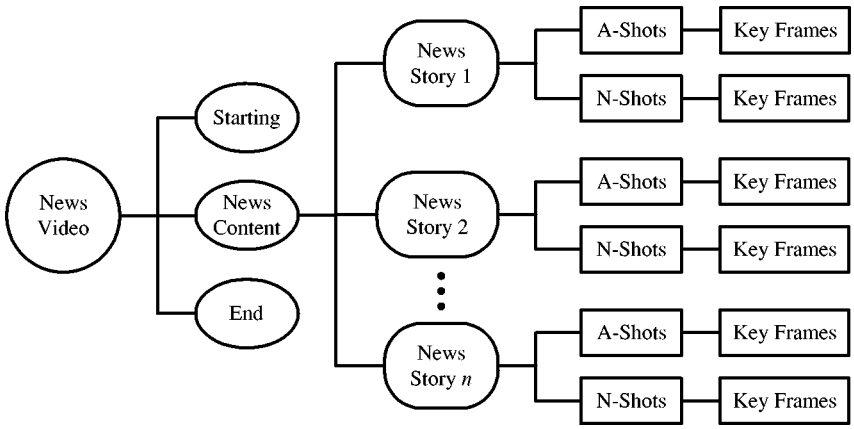


Fig. 1. Tree-structured representation of a typical news program. A-Shots: Anchorperson shots. N-Shots: News footage shots.

to much more effective retrieval than simple linear or keyword only search. Merlino *et al.* [27] empirically demonstrate that the speed with which a user can retrieve relevant stories that are well segmented can be orders of magnitude faster than the speed of linear search or simple keyword-based search. Without good story parsing, all other components of a video library are significantly less useful, because the user cannot locate desired material efficiently [20].

The MEDUSA system does not provide story parsing, while BNN uses simple linguistic discourse structure for segmentation. Although Informedia tries to integrate multiple cues from speech, closed caption, and visual content, it relies mostly on textual and linguistic information from closed-caption tokens and automatically recognized speech transcripts. Visual cues are limited to simple processing such as black screen detection and color histogram analysis and the overall segmentation performance is still less than satisfactory. As video's primary distinction from speech and text information, visual information in video should understandably play an important role in the story segmentation process. To effectively utilize visual cues, several techniques have been proposed [3], [11], [15], [18], [22], [45], [46]. Visual cues are not only essential for videos where textual information is hard to extract, but are also more stable than linguistic cues.

In [45], based on the observation of news programs aired by the Singapore Broadcasting Corporation (SBC), Zhang *et al.* assume a straightforward temporal syntax of a news video—a sequence of news stories interleaved with commercials. Each news story is composed of an anchorperson shot followed by relevant news footage. Therefore, a news program can be represented as a hierarchical tree-structured model as shown in Fig. 1. Such a simple news structure has been observed for news programs in Singapore (SBC) [22], [45], Hong Kong (ATV, TVB) [12], [26], and local news in the U.S. [15]. News story parsing, therefore, becomes a problem of how to distinguish anchorperson shots from news footage shots. Even for more complicated news program structures, anchorperson shots still serve as the root shots for constructing news stories [20]. Thus, detecting anchorperson shots plays a key role in news story parsing.

Most of the existing anchorperson detection methods are based on the model matching strategy [3], [11], [15], [45]. Following Swanberg's proposal [34], Zhang *et al.* construct three models for an anchorperson shot: shot, frame, and region, [45]. An anchorperson shot is modeled as a sequence of frame models and a frame is modeled as a spatial arrangement of regions. Thus, recognizing an anchorperson shot involves testing that every frame satisfies a frame model, which in turn means testing each frame against a set of region models. These models vary for different TV stations. It is difficult to construct all the possible models for different news videos. Moreover, the model matching method has a high computational complexity. Gunsel *et al.* identify anchorperson shots by color classification and template matching [15]. They first extract possible regions where anchorpersons may be situated with skin detection and histogram intersection, then compare these regions with templates stored in the application data. Like Zhang's proposal, the creation and matching of templates are time-consuming processes and strongly depend on the application data. The face detection approach by Avrithis *et al.* [3] is quite complicated, with a number of parameters needing to be manually tuned. Given that face detection in a still image is already difficult enough, face detection in video is too time-consuming for practical application. Furthermore, since the method also requires training data for classification, it is not an unsupervised approach. The template based method by Hanjalic *et al.* [18] assumes that different anchorperson models have the same background. This is not true for most news stations. Because of different camera angles, different models have different backgrounds. Adding the changing clothes color and news icon, we find less than 30% similarity between different models in our dataset, which is not enough to distinguish the anchorperson shots. In addition, the complicated template matching method is very time consuming.

In this paper, we present a practical unsupervised shot boundary detection technique and a model-free anchorperson shot classification algorithm for news story parsing. First, a two-pass modified fuzzy c-means algorithm is used to detect the shot boundaries and partition the video frames into video shots. High-accuracy results are obtained through a fuzzy-membership based refinement process and the distinction between abrupt shot transition and gradual shot transition is found using a binary pattern matching method. Then, a graph-theoretical cluster (GTC) analysis method is employed to classify the video shots into anchorperson shots and news footage shots. The news story is finally constructed according

to the simple temporal syntax mentioned above. As a self-organized model-free approach, the proposed method is much simpler and faster than the model matching approach. We obtain high-accuracy experimental results on a large data set containing over 3,907 video shots and 255 news stories from the news programs of two leading news stations in Hong Kong. However, we need to point out that automatic parsing of all types of news story is not exactly achieved here. We are only using the simple news structure in Fig. 1 to test the efficacy of the two key components of a news parsing system.

The rest of this paper is organized as follows. First, the fuzzy clustering algorithm for shot boundary detection is introduced in Section II. Section III describes the GTC analysis method and its application in anchorperson shot identification. The experimental results are presented in Section IV. Finally, conclusions and suggestions for future work are given in Section V.

II. Shot Boundary Detection

A shot boundary corresponds to either a significant change in the overall color composition or a significant change in the object location or both. To reliably detect the frame-to-frame content change, we use two conventional frame difference metrics, the histogram difference metric (HDM) and the spatial difference metric (SDM), to measure the dissimilarity between the adjoining frame pair.

A. Definition of Dissimilarity Metrics

Let f_t denote the tth frame, and f_{t+1} denote the $(t+1)$th frame in a video sequence. Let $I_t(i,j)$ and $I_{t+1}(i,j)$ denote the intensity of a pixel at location (i,j) in the frames f_t and f_{t+1} respectively. Let the frame size be $M \times N$. The spatial difference based metric SDM is defined as

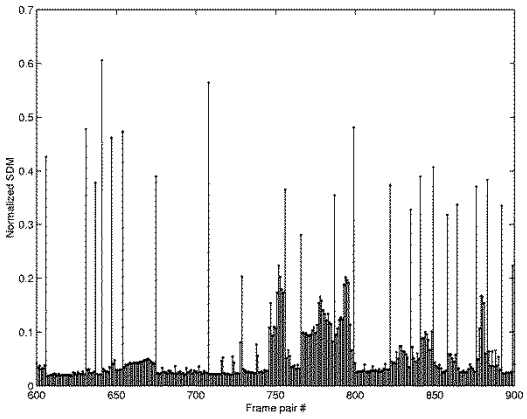
$$D_S(t) = \frac{1}{M \times N} \left(\sum_{i=1}^{M} \sum_{j=1}^{N} |I_t(i,j) - I_{t+1}(i,j)|^p \right)^{1/p} \quad (1)$$

where $p \in [1, +\infty)$. The cases of $p = 1$ and $p = 2$ are of particular interest. They are often called "city block" and "Euclidean" distances, respectively. Since the SDM does not take into account the camera motion, it may produce false alarms for shot boundary detection. To this end, we introduce another popular metric, histogram difference metric.

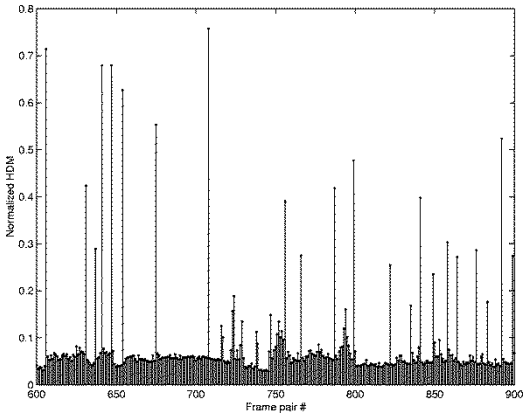
Let $H_t(k)$ denote the gray-level or color histogram for the tth frame, where k is one of the L possible colors or gray levels. Then the histogram difference based metric HDM is computed for every frame pair as

$$D_H(t) = \frac{1}{M \times N} \left(\sum_{k=1}^{L} |H_t(k) - H_{t+1}(k)|^p \right)^{1/p}. \quad (2)$$

For convenience of processing, we first normalize the SDM and HDM into the interval [0, 1]. Fig. 2 shows a plot of the normalized SDM and HDM (with $p = 1$) for a news video clip. If we slide a small window through the plot, we see that sometimes there is a single high pulse in the window and sometimes there are a number of consecutive medium-height pulses. They correspond to the two types of camera shot transition, break and dissolve, respectively. A break is an abrupt transition between two camera shots that is completed within two consecutive frames. A dissolve is a gradual transition between two camera-shots that takes several frames and can be a fade-out, a fade-in, or a combination of both.



(a)



(b)

Fig. 2. Plot of (a) normalized spatial difference metric and (b) histogram difference metric with the parameter $p = 1$.

Based on the frame difference metrics, traditional methods usually use a threshold to detect the shot boundaries. However, past studies on the statistical behavior of frame difference show that a threshold that is appropriate for one type of video data may not yield acceptable results for another type of input [15]. To overcome this major drawback, the K-means algorithm is proposed for unsupervised shot boundary detection [15], [30]. This method treats syntactic video segmentation as a simple two-class clustering problem, where the two-class labels are "significant change" (SC) and "no significant change" (NSC). In practice, there is no distinct boundary between the two categories, since the distinction between the features of a shot boundary frame and a regular frame is rather ambiguous. So, we introduce the fuzzy classification technique to distinguish the two categories of frame pairs.

B. Fuzzy Classification

After extracting the frame difference metrics from the video data, all frame pairs are mapped to a point-set in the feature space F_D spanned by the SDM and HDM metrics

$$F_D = \left\{ F_D(t) = (D_S(t), D_H(t)) \,\middle|\, t = 1, 2, \ldots, T \right\}. \quad (3)$$

Then, the problem of shot boundary detection becomes partitioning the feature space into two subspaces, SC and NSC.

First, we define the fuzzy 2-partition of the feature space F_D. Let V_{2T} be a set of real $2 \times T$ matrices and u_{it}, $i = 1, 2$, $t = 1, 2, \ldots, T$ denote the membership degree of the tth vector $F_D(t)$ in the i-th category. Then the fuzzy 2-partition space of F_D is defined by

$$M_{f2} = \left\{ U \in V^{2T} \middle| u_{it} \in [0,1], \forall i, t; \sum_{i=1}^{2} u_{it} = 1, \forall t; 0 < \sum_{t=1}^{T} u_{it} < T, \forall i \right\}. \quad (4)$$

The two rows of the matrix $U \in M_{f2}$ define two fuzzy subsets corresponding to the SC and NSC subspaces.

To obtain the optimal c fuzzy subspaces, we adopt the fuzzy c-means (FCM) algorithm. FCM is an objective function based fuzzy clustering algorithm. The least-mean-square error is often used as the criterion function. For our application, the objective function can be constructed as

$$J_m(U, v) = \sum_{i=1}^{c} \sum_{t=1}^{T} (u_{it})^m \cdot \|F_D(t) - v(i)\|^2 \quad (5)$$

where $c = 2$ corresponds to the fuzzy 2-partition, $U \in M_{f2}$, $v \in R^{2\times 2}$ is the cluster center or prototype of the fuzzy subset $\{u_i | i = 1, 2\}$, and $m \in [1, +\infty)$ is the weight exponent, which controls the fuzziness of the algorithm. In general, the proper range for m is [1.5, 2.5]. Here, we take $m = 2$ in the FCM algorithm.

By minimizing the objective function (5), we obtain the optimal fuzzy space partition U^* and the optimal cluster prototype v^* [5], given by

$$u_{it}^* = \left(\sum_{k=1}^{c} \left(\frac{\|F_D(t) - v^*(i)\|}{\|F_D(t) - v^*(k)\|} \right)^{2/(m-2)} \right)^{-1} \quad (6)$$

$$v^*(i) = \frac{\sum_{i=1}^{T} (u_{it}^*)^m \cdot F_D(t)}{\sum_{t=1}^{T} (u_{it}^*)^m}. \quad (7)$$

Fig. 3 shows a plot of feature points in the 2-D metric space and the partition result using the FCM algorithm. The final classification result is obtained by defuzzifying the optimal fuzzy partition matrix U^*. This operation contains two steps. First, for the $F_D(t)$ with a distinctive membership value, *i.e.*, $u_{it} \in [0, 0.4] \cup [0.6, 1]$, the defuzzification is carried out according to the rule

$$\mu(t) = \begin{cases} 1: & u_{1t} \geq u_{2t} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $u_1 = \{u_{1t} | t = 1, 2, \ldots, T\}$ denotes the fuzzy subspace of the SC category, u_2 denotes the fuzzy subspace of the NSC category and $\mu(t)$ is the indicator function of the feature vector $F_D(t)$ in the SC category.

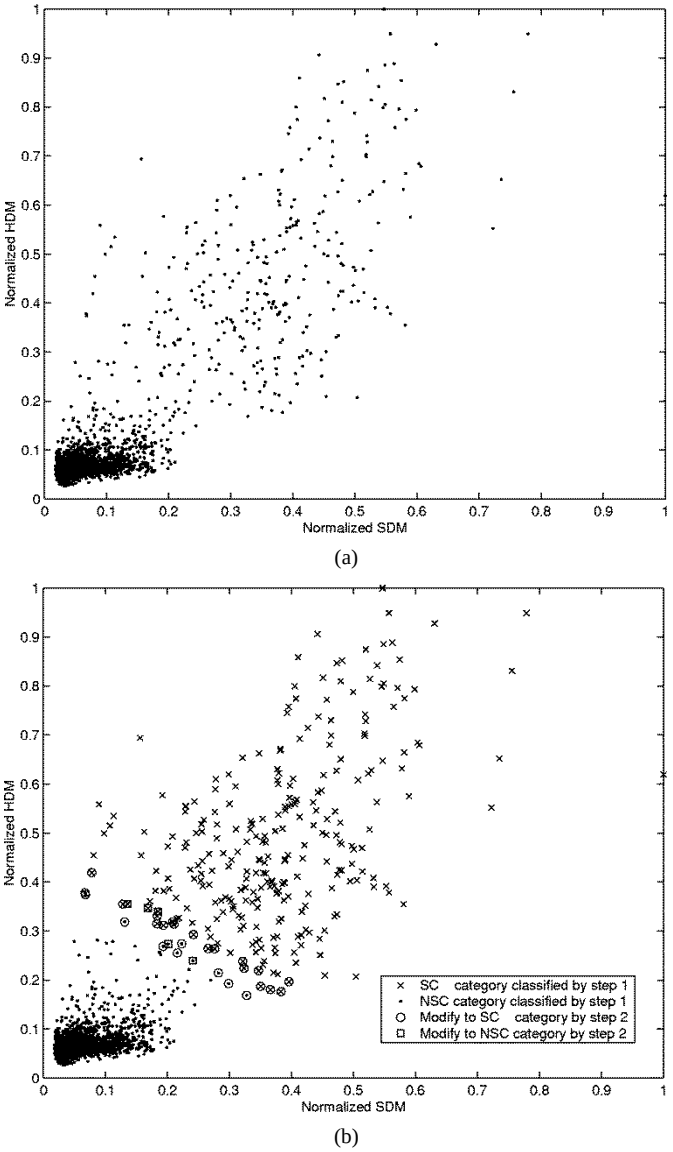


Fig. 3. (a) Frame pairs in the feature space. (b) Classification results using the two-step FCM algorithm.

For the $F_D(t)$ with a fairly fuzzy membership value, *i.e.*, $u_{it} \in [0.4, 0.6]$, we further refine the classification of $F_D(t)$. We define a new frame difference metric

$$D_F(t) = D_S(t) \times D_H(t). \quad (9)$$

For the $F_D(t)$ with $u_{it} \in [0.4, 0.6]$, if the corresponding $D_F(t)$ is a local maxima, then at least one of $D_S(t)$ and $D_H(t)$ is a local maxima. For a frame pair with a larger frame difference metric than all its neighbor frame pairs, it more likely corresponds to a shot boundary. Therefore, we classify those fuzzy frame pairs with local maxima $D_F(t)$ into the SC category. Using the two-step defuzzification scheme, we obtain the crisp partition result of the frame pairs shown in Fig. 3. In the plot, the points whose membership results are modified by the second step processing are denoted by "$\odot$" (change from NSC to SC) and "$\boxtimes$" (change from SC to NSC).

Fig. 4. Key frames of news footage shots in a news program.

C. Two-Pass Shot Boundary Detection

Since the difference metrics SDM and HDM normally have to be computed for every contiguous frame pair in the video stream, for a 20-min news program there are approximate 30 000 (20×60 s $\times$ 25 fps) frame pairs to be processed. The computational cost is fairly high. Moreover, too many frame pairs from the NSC category will degrade the performance of the FCM algorithm since the algorithm usually performs best when the two categories are comparable in size. To reduce the computational complexity, we propose a two-pass scheme to speed up the shot boundary detection process.

In the first pass, we only compute frame difference between two frames that are a fixed interval apart. Assume the interval is set at $\triangle t$, then the frame difference metrics are extracted only for frame pair $(f_{n\triangle t}, f_{(n+1)\triangle t})$. Thus, for a program of length T, there are totally $T/\triangle t$ operations in the first step. In the second pass, we only need to conduct frame by frame computation on those intervals where significant change are detected in order to find the accurate shot location. Assume that S shot boundaries are detected in the first step, then the second pass will need $\triangle t \cdot S$ operations. To minimize the total number of operations in the two passes

$$\min\left\{\frac{T}{\triangle t} + \triangle t \cdot S\right\}$$

we need to have $\triangle t = \sqrt{T/S}$. By manually labeling the shot boundaries of large amount of news programs, we find that the number of shot boundaries for a 20-minute news program is around 1% of the total number of video frames, *i.e.*, $S = 1\% \cdot T$. Therefore, the optimal interval $\triangle t$ approximates 10, which gives the total number of operations as 6000, or only 20% of the original operations.

As mentioned above, there exist two types of video-shot transition, abrupt and gradual transition. For the abrupt transition, there is only one single frame pair with significant change of the frame difference metrics. However, for the gradual transition, there are a number of consecutive frame pairs with significant content changes. Here we treat each gradual transition as a very short camera shot. We only consider the first and last frame pairs of a gradual transition as shot boundaries. So, not all the frame pairs in the SC category are shot boundaries. To locate the shot boundaries, we analyze the "0–1" binary string formed by the sequence of $\mu(t)$, $t = 1, 2, \ldots, T$. An abrupt transition corresponds to the sequence pattern, "010" and the boundary of a gradual transition corresponds to the pattern "011" or "110". By detecting the three sequence patterns, we can locate the exact location of shot boundaries.

III. Video-Shot Classification

Once a news video is partitioned into individual video shots, the next step of video segmentation is to classify the video shots into anchorperson shots and news footage shots.

For convenience of processing, one frame is extracted from each video shot as a key frame to represent the shot. Most traditional methods use model matching on these key frames to identify the anchorperson shots. However, the template model building process is complex and time-consuming. To develop a model-free method, we introduce the GTC algorithm to analyze the relationship among the key frames. We know that each single news program has a finite number of models for the anchorperson frames and each anchorperson model appears no less than twice in a news program. We also observe that most key frames of news footage shots are very different from each other as shown in Fig. 4. However, the key frames from two anchorperson shots of the same model are very similar. Fig. 5 shows examples of key frames of anchorperson shots in a Hong Kong TVB station news program (January 26, 2000). We can see that since the key frames with the identical model have the same background and anchorperson, they thus have similar color histogram and spatial content. Based on this similarity, we can group anchorperson shots of each model in a self-organized fashion through the GTC method to distinguish them from the individually distributed news footage shots.

A. GTC Analysis

The GTC method was initiated by Zahn [43]. It takes the given data as vertices in the feature space and then constructs the minimum spanning tree on these vertices. Cutting those edges

Fig. 5. Key frames of anchorperson shots in a news program.

greater than a given threshold, it explores the proximity relationship of nodes using connectivity among the nodes. We discuss the detailed algorithm in this section.

First, several terms on the graph theory and a theorem on the minimum spanning tree need to be reviewed [4].

Graph: A nonorientated graph G is a set of *vertices* (*nodes*) and *edges* (*arcs*) which connect them

$$\begin{aligned} G =& [X, E] \\ X =& \{x_1, x_2, \ldots, x_n\} \\ E =& \left\{e_{ij} = (x_i, x_j) \;\middle|\; x_i, x_j \in X\right\} \end{aligned} \tag{10}$$

where X is the set of vertices and E is the set of edges.

Path: A path P of length K through a graph is a sequence of connected vertices: $P = \langle x_1, x_2, \ldots, x_{K+1}\rangle$ where, for all $i \in (1, K)$, (x_i, x_{i+1}) is in E.

Cycle: A graph contains no cycles if there is no path of nonzero length through the graph $P = \langle x_1, x_2, \ldots, x_{k+1}\rangle$, such that $x_1 = x_{k+1}$.

Spanning Tree: A spanning tree of a graph G is a set of $(n-1)$ edges that connect all vertices of the graph. A tree is a graph $[X, T]$ without a cycle. The graph $[X, T]$ is a tree if and only if there exists one and only one path between any pair of vertices.

Minimum Spanning Tree (MST): In general, it is possible to construct multiple spanning trees $[X, T_i]$ $(i > 1)$ for a graph G. If a weight $w(e)$ is associated with each edge e then the minimum spanning tree is the set of edges, MST, forming a spanning tree, such that

$$w(\text{MST}) = \min_i \left\{ \sum_{e \in T_i} w(e) \right\}. \tag{11}$$

It is unique if all weights are different.

Cocycle: By removing an edge e from the tree $[X, T]$, we create two connected components of vertices $A \subset X$ and $\bar{A} = X - A$. The cocycle θ^e is defined as:

$$\theta^e = \left\{e_{ij} \;\middle|\; x_i \in A, x_j \in \bar{A}, \bar{A} = X - A\right\} \tag{12}$$

i.e., the set of edges that connect a vertex of A with a vertex of $\bar{A}$ in graph $[X, G]$.

The following theorem gives the necessary and sufficient condition for constructing a minimum spanning tree.

Theorem 1: A necessary and sufficient condition for $[X, T]$ to be a minimum spanning tree of G is that for all edges $e \in E$ the cocycle θ^e (such that $\theta^e \cap T = \{e\}$) satisfies: $w(e) \leq w(s), \forall s \in \theta^e (s \neq e)$.

Forest: A graph without a cycle, and which is not connected, is called a *forest*. Each connected component is a *tree*.

For the GTC analysis, we can use the feature vectors of the studied objects as the vertices of a nonoriented graph $G = [X, E]$. So for a set of n objects, we have a vertex set $X = \{x_1, x_2, \ldots, x_n\} \subset R^p$. We define the weight of any edge $e_{ij}, (1 \leq i, j \leq n)$ as the distance between the node pair (x_i, x_j)

$$w(e_{ij}) = \|x_i - x_j\|, \; x_i, x_j \in X. \tag{13}$$

We can then group the object clusters through the following steps.

1) Construct the minimum spanning tree using the Prim algorithm [4]

$$\text{MST} = \left\{(A, T) \;\middle|\; A = X, T = \{e_1, \ldots, e_{n-1}\}\right\}$$

with

$$w(\text{MST}) = \min\left\{w(\text{Tree}) \;\middle|\; \text{Tree} = (X, T')\right\}. \tag{14}$$

2) Cut the edges whose weights exceed a threshold γ from the MST to form a forest on the node set X

$$F = \left\{(X, E') \;\middle|\; E' = T - \{e' | w(e') > \gamma\}\right\}. \tag{15}$$

3) Find all the trees $\{(X_i, T_i) | i = 1, 2, \ldots, m\}$ contained in the forest F

$$F = \bigcup_{i=1}^{m} (X_i, T_i), \text{where} \bigcup_{i=1}^{m} X_i = X, \bigcup_{i=1}^{m} T_i = E'. \tag{16}$$

4) Consider each tree (X_i, T_i) as a potential object cluster.

To demonstrate the GTC analysis, we give a simple example in Fig. 6. For a point-set of 40 nodes in a 2-D space, the city block distance between any two nodes is used to define the weight of the edge. Using the Prim algorithm, a minimum spanning tree is obtained as shown in Fig. 6(a). By removing all the edges with weights greater than a threshold ($\gamma = 1$), we arrive at a forest containing six trees (clusters), which are the connected components shown in Fig. 6(b). From this example, we see that the GTC method automatically groups similar nodes

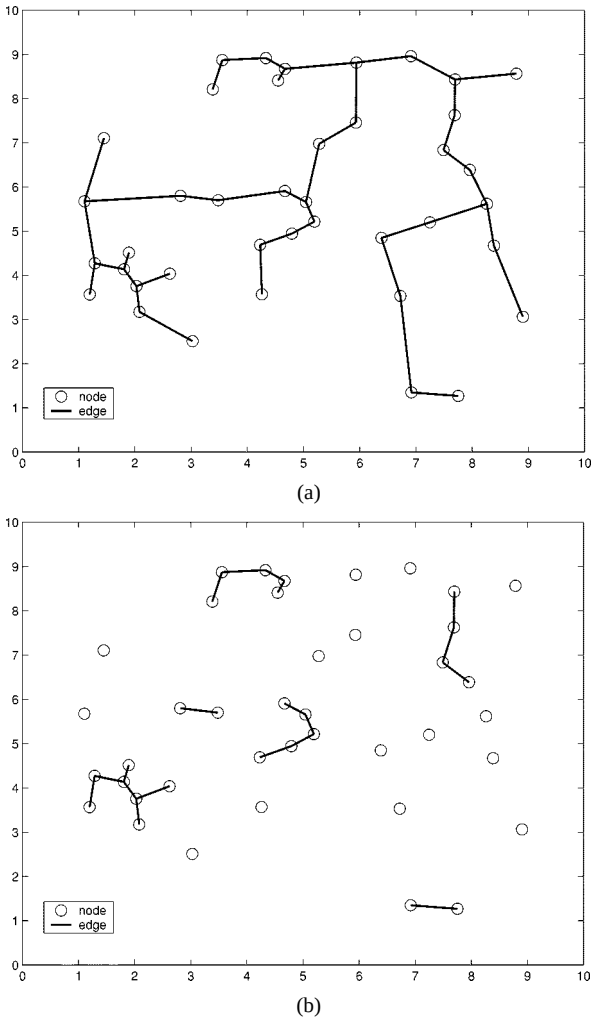


Fig. 6. (a) MST. (b) Forest obtained by the GTC algorithm.

into clusters. Since each anchorperson model appears several times in a news program and the key frames in different shots have very similar content, we can use GTC analysis to group similar anchorperson frames into clusters to separate them from key frames of news footage shots.

B. Anchorperson Shot Detection

As shown in Fig. 7, the anchorperson shot detection scheme is composed of four steps: short shot filtering, key frame extraction, GTC analysis and post-processing.

In general, an anchorperson shot should last for more than 2 s, since the shot should involve at least one sentence by the reporter. Therefore, given the total number of frames N_S^i in a shot, assuming the playing frame rate of the news program is R_f, if $N_S^i < 2 \cdot R_f$, the shot is considered as news footage. Otherwise, the shot will be further analyzed through later steps. This helps to reduce the computational burden of the following modules.

The next step is the key frame extraction. In general, key frame selection is itself an important research topic. Selecting the most representative key frame is important for many video processing applications. This is not quite the case for the anchorperson detection study. Since our focus is on anchor shot detection, key frame selection in an anchorperson shot affect the algorithm performance much more than key frame selection in a scene shot. However, for an anchorperson shot, video frames remain almost stationary, thus choosing which frame to be a key frame does not make much of a difference. On the other hand, for a scene shot, key frame selection may be important for such application as visual abstraction, but hardly makes a difference in anchorperson detection. Since no matter which frame is selected, as long as it is not identical to key frames from other video shots, it will not be identified as an anchorperson candidate. Considering possible camera motions at the beginning and end of a shot, we simply take the middle frame as the key frame. These key frames are the input to the GTC analysis module.

Since we need to construct a nonoriented weighted graph on the key frames in the cluster analysis module, we first define the weight of an edge in the graph as the distance between two key frames. To be robust to noise, the metric proposed in [28] is used to compute the distance between two key frames. As shown in Fig. 8, the two key frames of Fig. 8(a) are divided into 16 regions of the same size. Then, histograms of corresponding regions in the two key frames are compared and the eight regions with the largest histogram differences are discarded to reduce the effects of object motion and noise. The remaining regions of the two key frames are shown in Fig. 8(b). The distance between these two key frames is then defined as the sum of the histogram differences of the remaining regions. Since we use the color histogram of the 16 regions $H_j^i(k)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, 16$, $k = 1, 2, \ldots, L$ to describe the key frames, where n is the number of key frames and L is the number of colors, the key frames $\{H^i | i = 1, 2, \ldots, n\}$ can be considered as a set of points in the space $R^{16 \times L}$. So the GTC algorithm can be used to analyze the proximity relationship of the key frames. Since the anchorperson key frames of the same model have very similar color histograms, they will be grouped into subtrees in the key frame forest. The computational complexity of the algorithm is not very high. Given N key frames in a news program, only $N(N-1)/2$ vector distances need to be computed.

For the output of the GTC algorithm, we consider all the key frames in a subtree with no less than two nodes as potential anchorperson frames, i.e.,

$$A_{p1} = \left\{ x \in \bigcup_i T_i \;\middle|\; T_i \in \text{MST and } |T_i| \geq 2 \right\} \qquad (17)$$

where MST is the constructed minimum spanning tree on the key frames, T_i is the obtained subtree by the GTC algorithm and $|T_i|$ is the size or node number of T_i.

It is possible that some news footage shots also have nearly identical key frames, which may be grouped into a cluster or subtree by the GTC algorithm. For example, a person giving a speech is shown several times in a news story. Fortunately, most of these types of news footage shots appear in a single news story. When the false anchorperson shots appear several times in one story, they cut the story into several small pieces. By *a priori* knowledge, we know that a complete news story should last at least 10 s. Therefore, a minimal interval filtering can be used to detect false anchorperson clusters. After the minimal interval of each cluster is computed, we remove the key frame

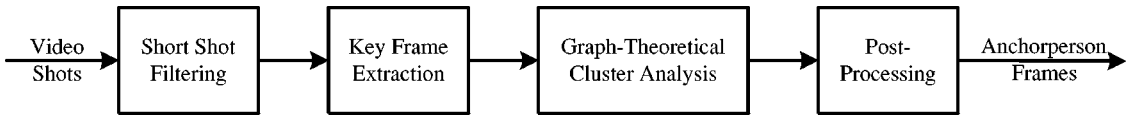


Fig. 7. Diagram of the anchorperson shot detection scheme.

(a)

(b)

Fig. 8. (a) Two key frames were divided into 16 regions. (b) Eight regions of the two key frames with similar histograms.

clusters that contain a news story shorter than 10 s. Finally, we get the refined potential anchorperson frames

$$A_{p2} = \left\{ A_{p1} - T_i \;\middle|\; \exists \triangle t(x_i, x_j) < 10s,\ \forall\, x_i, x_j \in T_i \right\}. \tag{18}$$

In some situations, the key frames in a cluster may have similar histograms but different content. For example, two frames may have the same background and objects, but the objects are at different locations in the frames. To detect this situation, we use the spatial difference metric SDM between the frames in a cluster. If a cluster has an average $\overline{\text{SDM}}$ higher than a threshold λ, the whole cluster is removed from the anchorperson frame list. By using the SDM filtering, we obtain the final anchorperson frames

$$A_{p3} = \left\{ A_{p2} - T_i \;\middle|\; \overline{\text{SDM}}\,(T_i) > \lambda \right\}. \tag{19}$$

In the anchorperson shot detection scheme, two thresholds, the γ for the GTC algorithm and the λ for the post-processing, need to be specified in advance. In general, they can be estimated by the fuzzy c-means algorithm. In the syntactic segmentation step, the FCM algorithm produces an optimal cluster prototype v^*. The prototype of NSC category $v^*(2)$ is a 2-D vector (D_S^*, D_H^*). D_H^* and D_S^* correspond to the average SDM metric and average HDM metric respectively. Both γ and λ are positively proportional to the D_H^* and D_S^* respectively. So, the selection of these thresholds can be computed adaptively based on the result of the FCM algorithm.

IV. Experimental Results

The methodology described above is applied to 14 news programs from the TVB and ATV news stations in Hong Kong. The detailed information of these news videos is summarized in Table I.

To evaluate the performance of the proposed scheme of news story segmentation, we use the standard *precision* and *recall* criteria, shown in the following:

$$\text{precision} = \frac{\text{number of hits}}{\text{number of hits} + \text{number of false alarms}} \tag{20}$$

$$\text{recall} = \frac{\text{number of hits}}{\text{number of hits} + \text{number of misses}} \tag{21}$$

A. Shot Boundary Detection Experiment

First, we evaluate the performance of the shot boundary detection algorithm. Table II shows the output of each step in the shot boundary detection process for the 14 news videos. For the total 3,907 video shots, there are 3,893 shot boundaries. The FCM algorithm detects 3,746 shot boundaries. The defuzzification post-processing further refines the results, adding 120 missed shot boundaries and excluding 36 false boundaries. Overall, the proposed method detects 3,830 video-shot boundaries, including 3,756 real shot boundaries and 74 false alarms. Thus, the precision is 98.07% and the recall is 96.48%.

In the final shot boundary set, there are 3,468 abrupt transitions and 362 gradual transmissions. Although the gradual transitions constitute only a minor portion of the overall shot transitions, most of the misses and false alarms of the shot boundaries appear with them. The misses are mainly due to the small content change between the frame pairs at some shot boundaries and the false alarms mostly result from the irregular camera operations during the gradual transitions.

The results seem much better than the state of the art algorithms evaluated in [13], where the best algorithm achieves 90%–95% recall with 70%–80% precision for abrupt transitions on 76 min of video programs. Our results are comparable to the results in [30], where a recall rate of 94.21% is achieved at 98.5% precision for 190 shot boundaries. However, our algorithms are much faster because of the two new techniques; the two-pass shot boundary detection and the selective refinement based on fuzzy membership values. We also distinguish between gradual and abrupt transition using a novel binary pattern-matching scheme.

B. Anchorperson Detection Experiment

The experimental results of anchorperson detection are given in Table III. Based on the 3,830 shot boundaries detected, all the news video programs are partitioned into video shots. After filtering out the too-short shots, we obtain 2,876 key frames for

TABLE I
DETAILED INFORMATION OF THE NEWS VIDEO PROGRAMS

News video	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Duration	21:29	21:46	21:49	27:05	22:02	22:18	22:49	20:22	21:49	21:08	20:29	20:40	20:55	20:36
Video shots	290	279	297	311	292	250	295	239	285	272	245	227	344	281
News stories	22	21	18	19	20	15	20	14	17	17	18	19	17	18
File type	MPEG-1							MPEG-1						
Frame size	288×352 pixel							288×352 pixel						
Frame Rate	25 frames/second							25 frames/second						
Source	TVB station							ATV station						

TABLE II
SHOT BOUNDARY DETECTION ALGORITHM RESULTS

News program video		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Detected by FCM algorithm		277	267	291	307	277	231	284	224	281	256	234	207	338	272
Corrected by defuzzification	Adding	9	7	6	7	17	9	8	5	5	9	5	10	12	11
	Canceling	3	2	2	5	5	0	2	1	3	0	1	3	6	3
Final results	Hits	278	269	289	304	281	237	286	223	276	261	231	211	336	274
	Misses	11	9	7	6	10	12	8	15	8	10	13	15	7	6
	False alarms	5	3	6	5	8	3	4	5	7	4	7	3	8	6

TABLE III
RESULTS OF ANCHORPERSON SHOT DETECTION ALGORITHM

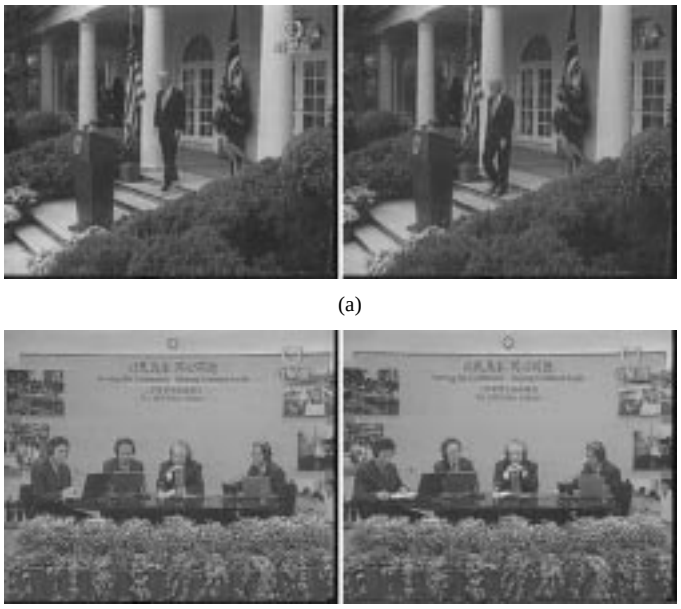
News program video		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Output of short shot filter		204	200	221	242	203	216	213	191	210	225	162	182	204	203
Detected by GTC algorithm		34	36	33	26	25	29	30	28	23	23	30	28	41	37
Output of post-processing		22	21	18	19	18	16	22	14	17	17	18	19	17	16
Final results	Hits	22	21	18	19	18	14	20	12	17	17	18	19	17	16
	Misses	0	0	0	0	2	1	2	0	0	0	0	0	0	2
	False alarms	0	0	0	0	0	2	2	2	0	0	0	0	0	0

the 14 news programs. Then, the GTC algorithm is used to analyze the key frames of each individual news program and identifies 423 potential anchorperson key frames. Post-processing refines the result and finally find 254 anchorperson key frames, in which we hit 248 real anchorperson key frames and get six false alarms. Therefore, we achieve a precision of 97.64% and recall of 97.25% for anchorperson shot detection. Note that because of the high accuracy in the shot boundary detection step, no error in the anchorperson detection step is caused by the mistakes in shot boundary detection step.

We find two types of errors in our experiments. The first type of error is due to the fact that some anchorperson models appear only once in a complete news program. Fig. 9(a)–(c) show the missed anchorperson key frames in news videos 5, 6, and 7, respectively. In these news programs, the anchorperson appears at the right/left side of frames in most cases. Since the missed frame models appear only once in the program, they are impossible to be detected by the cluster method. Fortunately, although the models of the missed frames appear only once in a single news program, they may appear several times in a combi-

(a) (b) (c)

Fig. 9. (a) Two missed anchorperson shot key frames in news video 5. (b) Missed anchorperson shot key frame in news video 6. (c) Two missed anchorperson shot key frames in news video 7.



(a) (b)

Fig. 10. (a) Two false key frames of anchorperson shots in news video 6. (b) Two false key frames of anchorperson shots in news video 8.

nation of a few news programs from the same news station. So, it is possible to reduce the miss rate of anchorperson frames by applying the GTC algorithm to a combination of more than one news program. For example, if videos 5, 6, and 7 are processed together, the first four of the five missing key frames would be detected because they appear twice in the overall program.

The second type of error is caused by news footage shots that are identical to each other. Fig. 10 shows two pairs of news shot key frames which are falsely identified as anchorperson key frames. The two key frames shown in Fig. 10(a) are the frames 143 97 and 245 60 frames and appear in the seventh and tenth news stories of news program 6, respectively. The two frames shown in Fig. 10(b) are frames 1299 and 7975, respectively, which appear in the first and third news stories of news program 8, respectively. Thus, not only do the two key frames look nearly identical, they also appear in two different news stories. This happens only in the rare situation when a news station uses identical footage for different news stories in the same day. In our test data, it only happens twice in the 14 days of news broadcasts.

Our post-processing step failed to detect these errors. This type of error may also be removed by integrating several news programs together and increasing the minimum key-frame number requirement in a cluster to more than two. Though we need to be cautious in deciding the number of news programs to integrate, with care given to both miss and false alarm errors. Since we have limited the false alarms to a very low rate, we can also afford to use more complicated method, such as anchorperson face recognition, to remove the errors.

Compared to existing model-based algorithms, our approach not only gives better performance, but is also computationally much simpler and requires little human intervention. The face detection approach by Avrithis *et al.* [3] achieves 97% recall at precision 95% in a test on 60 min of news video. The algorithm is complicated and computation intensive. In [15], the skin color detection algorithm correctly identified 142 out of 147 shots. Note that the 147 shots are the total number of shots, including both the anchorperson and news footage shots. In our case, we only incorrectly identified 13 out of more than 3,000 shots. In [22], with a set of predefined anchorperson models for two news programs of 39 stories, 38 stories are detected with four false alarms. None of the existing research reports computational complexity and only a small data set is used for their tests. The computational complexity of the GTC algorithm is $O(n^2)$ with respect to the number of key frames in a news program. The robust and high-accuracy performance of our algorithm are clearly shown in the detailed experiments on a much larger data set (to the best of our knowledge, the dataset is the largest so far).

V. Conclusion and Future Work

We present an unsupervised video-shot segmentation method and a model-free anchorperson detection scheme for news story parsing. The system first segments the news program into video shots with the fuzzy c-means clustering algorithm. We introduce several novel techniques to improve the shot segmentation performance, including the two-pass strategy, postprocessing based on fuzzy membership values and abrupt and gradual transition differentiation through binary pattern matching. After the news video is parsed into camera shots, we detect anchorperson shots using the GTC algorithm. Individual news stories can then be constructed based on a much simplified temporal structural model of the news program. Experimental results on a data set significantly larger than most news video experiments in the literature have shown that both the shot boundary detection method and the anchorperson detection method are highly efficient.

We did not address the commercial breaks and the starting and ending sequence in the news video, since they can be detected

by the existing work [20], [36]. Since the proposed scheme depends on a much simplified temporal structural model of news video, it has some inherent drawbacks. For instance, it cannot identify a change of news items within a single anchor shot sequence. It is impossible to overcome such a drawback using visual information alone. In many situations, news programs do not follow the simple temporal structure described here, therefore text captions and speech signals have to be combined with the visual information to segment news stories. We are working on integrating this work with text and speech-based news video analysis methods to develop a robust news story segmentation system.

ACKNOWLEDGMENT

The authors thank Hong Kong TVB and ATV stations for providing the news programs, and also Dr. Q. Yang for valuable comments.

REFERENCES

[1] A. Akutsu, "Video indexing using motion vectors," *Proc. SPIE Visual Communications and Image Processing*, vol. SPIE 1818, pp. 1522–1530, 1992.

[2] Y. A. Aslandogan and C. T. Yu, "Techniques and systems for image and video retrieval," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 56–63, 1999.

[3] Y. Avrithis, N. Tsapatsoulis, and S. Kollias, "Broadcast news parsing using visual cues: A robust face detection approach," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, vol. 3, 2000, pp. 1469–1472.

[4] R. Balakrishnan and K. Ranganathan, *A Textbook of Graph Theory*. New York: Springer-Verlag, 1999.

[5] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithm*. New York: Plenum, 1981.

[6] J. S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," in *Proc. IS&T/SPIE Conf. Storage and Retrieval for Image and Video Databases IV*, vol. SPIE 2670, 1996, pp. 170–179.

[7] P. Bouthemy, M. Gelgon, and F. Ganansia, "A unified approach to shot change detection and camera motion characterization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 1030–1044, 1999.

[8] S. Boykin and A. Merlino, "Improving broadcast news segmentation processing," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, vol. 1, 1999, pp. 744–749.

[9] M. G. Brown, J. T. Foote, G. J. F. Jones, K. S. Jones, and S. J. Young, "Automatic content-based retrieval of broadcast news," in *ACM Multimedia'95*, San Francisco, CA, 1995, pp. 35–43.

[10] Y. L. Chang, W. Zeng, I. Kamel, and R. Alonso, "Integrated image and speech analysis for content-based video indexing," in *Proc. IEEE Multimedia'97*, 1996, pp. 306–313.

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

[12] X. Gao and X. Tang, "Automatic news video caption extraction and recognition," *Lecture Notes in Computer Science 1983*, pp. 425–430, 2000.

[13] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance characterization of video-shot-change detection method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1–13, 2000.

[14] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, and M. Sakauchi, "Automatic parsing of TV soccer programs," in *Proc. Int. Conf. Multimedia Computing and Systems*, May 1995, pp. 167–174.

[15] B. Gunsel, A. M. Ferman, and A. M. Tekalp, "Video indexing through integration of syntactic and semantic features," in *Proc. Workshop Applications of Computer Vision*, Sarasota, FL, 1996, pp. 90–95.

[16] A. Hampapur, R. Jain, and T. Weymouth, "Production model based digital video segmentation," *Multimedia Tools Applic.*, vol. 1, no. 1, pp. 9–46, 1995.

[17] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 580–588, 1999.

[18] ——, "Template-based detection of anchorperson shots in news programs," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Chicago, IL, 1998, pp. 148–152.

[19] A. G. Hauptmann and M. J. Witbrock, "Informedia: News-on-Demand multimedia information acquisition and retrieval," in *Intelligent Multimedia Information Retrieval*, M. T. Maybury, Ed. Menlo Park, CA: AAAI Press, 1997, pp. 213–239.

[20] ——, "Story segmentation and detection of commercials in broadcast news video," in *Proc. Advances in Digital Libraries Conf.*, Santa Barbara, CA, Apr. 1998, pp. 168–179.

[21] C. F. Lam and M. C. Lee, "Video segmentation using color difference histogram," in *Lecture Notes in Computer Science 1464*. New York: Springer Press, 1998, pp. 159–174.

[22] C. Y. Low, Q. Tian, and H. J. Zhang, "An automatic news video parsing, indexing and browsing system," in *Proc. ACM Multimedia*, Boston, MA, 1996, pp. 425–426.

[23] M. Maybury, A. Merlino, and J. Rayson, "Segmentation, content extraction and visualization of broadcast news video using multistream analysis," in *AAAI Spring Symposium*, Stanford, CA, 1997, pp. 1–12.

[24] M. T. Maybury and A. E. Merlino, "Multimedia summaries of broadcast news," in *Proc. Int. Conf. Intelligent Information Systems*, 1997, pp. 442–449.

[25] J. Meng, Y. Juan, and S. F. Chang, "Scene change detection in a MPEG compressed video sequence," in *Proc. SPIE/IS&T Symp. Electronic Imaging Science and Technologies: Digital Video Compression: Algorithms and Technologies*, vol. 2419, 1995.

[26] H. M. Meng, X. Tang, P. Y. Hui, X. Gao, and Y. C. Li, "Speech retrieval with video parsing for television news programs," in *Proc. ICASSP'2001*, Salt Lake City, UT, 2000.

[27] A. Merlino, D. Morey, and M. Maybury, "Broadcast news navigation using story segmentation," in *Proc. ACM Multimedia*, Nov. 1997, pp. 381–389.

[28] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-search for video appearances," in *Visual Database Systems*, E. Knuth and I. M. Wegener, Eds. Amsterdam, The Netherlands: Elsevier, 1992, vol. II, pp. 113–127.

[29] Y. Nakamura and T. Kanade, "Semantic analysis for video contents extraction—Spotting by association in news video," in *Proc. 5th ACM Int. Multimedia Conf.*, 1997.

[30] M. R. Naphade, R. Mehrotra, A. M. Ferman, J. Warnick, T. S. Huang, and A. M. Tekalp, "A high-performance shot boundary detection algorithm using multiple cues," in *Proc. Int. Conf. Image Processing*, vol. 1, 1998, pp. 884–887.

[31] N. V. Patel and I. K. Sethi, "Video shot detection and characterization for video databases," *Pattern Recognit.*, vol. 30, pp. 583–592, 1997.

[32] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *J. Vis. Commun. Image Repres.*, vol. 7, no. 4, pp. 345–353, December 1996.

[33] B. Shahraray, "Scene change detection and content-based sampling of video sequences," in *Proc. SPIE/IS&T Symp. Electronic Imaging Science and Technologies: Digital Video Compression: Algorithms and Technologies*, vol. 2419, 1995, pp. 2–13.

[34] D. Swanberg, C. F. Shu, and R. Jain, "Knowledge guided parsing and retrieval in video database," in *Proc. SPIE1908*, 1993, pp. 173–187.

[35] Y. P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 133–146, 2000.

[36] Y. Taniguichi, A. Akutsu, Y. Tonomura, and H. Hamada, "An intuitive and efficient access interface to real-time incoming video based on automatic indexing," in *ACM Multimedia*, San Francisco, CA, 1995, pp. 25–33.

[37] H. Wactlar, T. Kanade, M. Smith, and S. Stevens, "Intelligent access to digital video: Informedia project," *IEEE Trans. Comput.*, vol. 10, pp. 46–52, 1996.

[38] H. D. Wactlar, A. G. Hauptmann, and M. J. Witbrock, "Informedia: News-on-Demand experiments in speech recognition," in *Proc. ARPA Speech Recognition Workshop*, Harrimen, NY, 1996.

[39] H. D. Wactlar, "New directions in video information extraction and summarization," in *Proc. 10th DELOS Workshop*, Greece, 1999, pp. 1–10.

[40] ——, "Informedia—Search and summarization in the video medium," in *Proc. Imagina 2000 Conf.*, Monaco, 2000, pp. 1–10.

[41] B. L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 533–544, 1995.

[42] M. Yeung and B. L. Yeo, "Video content characterization and compaction for digital library applications," in *Proc. IS&T/SPIE Storage and Retrieval for Image and Video Databases V*, vol. 3022, Feb. 1997, pp. 45–58.

[43] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. 20, pp. 68–86, 1971.

[44] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.
[45] H. Zhang, Y. Gong, S. W. Smoliar, and S. Y. Tan, "Automatic parsing of news video," in *Proc. Int. Conf. Multimedia Computing and Systems*, 1994, pp. 45–54.
[46] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu, "Video parsing, retrieval and browsing: An integrated and content-based solution," in *Proc. Multimedia '95*, San Francisco, CA, pp. 15–24.
[47] H. J. Zhang, C. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools Applic.*, vol. 1, no. 1, pp. 91–113, 1995.

Xinbo Gao (M'01) received the B.S., M.S., and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively.

Since 1999, he has been with the School of Electronic Engineering, where he is currently an Associate Professor at the Department of Electronic Engineering. From 2000 to 2001, he was with the Department of Information Engineering, the Chinese University, Shatin, Hong Kong, as a Research Associate. His research interests include content-based video analysis and representation, image understanding, pattern recognition, and artificial intelligence.

Xiaoou Tang (S'93–M'96–SM'01) received the B.S. degree in 1990 from the University of Science and Technology of China, Hefei, China, and the M.S. degree in 1991 from the University of Rochester, Rochester, NY, and the Ph.D. degree in 1996 from Massachusetts Institute of Technology, Cambridge.

He is currently an Assistant Professor in the Department of Information Engineering of the Chinese University of Hong Kong, Shatin, Hong Kong. His research interests include video processing and pattern recognition.