Fuzzy framework for unsupervised video content characterization and shot classification

A. Müfit Ferman*

Sharp Labs of America Inc.
5750 NW Pacific Rim Boulevard
Camas, Washington 98607
E-mail: mferman@sharplabs.com

A. Murat Tekalp†

University of Rochester
Department of Electrical and Computer Engineering
Hopeman Hall 204
Rochester, New York 14627-0126

Abstract. In this paper we present a fuzzy framework for domain-dependent analysis of video sequences. Fuzzy clustering and cluster validation methods are first employed to determine the number of distinct shot patterns and construct a reference model for a program or video domain of interest, using an appropriate training set. This model is subsequently utilized to assign new input data to the available classes by a fuzzy minimum-distance classifier. Additional domain-specific information can be introduced after classification to further enhance the annotations associated with every shot. The main advantage of the approach is that it builds a model for the input video automatically from training data, and thus eliminates the need for extensive user supervision. The fuzzy representation method improves the interpretability of the results, and reduces the number of erroneous classifications, since the continuous class affiliations of each input sample provide a confidence measure for the final assignments. The proposed approach presents a computationally efficient, unsupervised method for building browsable semantic descriptions of video sequences. Specifically, the algorithm can be used to generate various components of an MPEG-7-compliant description. © 2001 SPIE and IS&T. [DOI: 10.1117/1.1406946]

1 Introduction

The ability to associate semantic descriptions with video data has emerged as a fundamental requirement for multimedia content management systems. Throughout the content production and consumption chain, involved parties (i.e., content creators, aggregators, providers, and end users) prefer to interact with the audio-visual (A/V) material using qualitatively descriptive metadata, rather than low-level signal features. This in turn necessitates annotation of A/V streams at different levels of semantic and spatio-temporal detail. The emerging multimedia standard

MPEG-7 (formally Multimedia Content Description Interface), scheduled for completion in October 2001,¹ aims to provide a complete and common set of tools for enabling both low-level and semantic characterization of A/V data, where the resulting content descriptions can be readily utilized and interpreted by a device or intelligent software agents, as well as human users. The standard is comprised of multiple visual and aural descriptors (Ds) that represent various low-level features;² multimedia description schemes (MDSs) that define the structure and organization of A/V content descriptions;³ and a description definition language⁴ that is based on the XML Schema Specification.⁵ MPEG-7 oversees the standardization of the (definitions of) description schemes and descriptors only; it does not, however, specify any normative methods for video sequence analysis and feature extraction. Overall, MPEG-7 offers a flexible and extensible framework that facilitates description of multimedia content at any selected granularity and level of abstraction, and simultaneously promotes interoperability.

Construction of semantic descriptions for video streams, like those enabled by MPEG-7, often requires efficient video analysis methods. Coarse-level information about video sequences, such as their genre or domain type, can be provided through user input, either during production or afterwards. As the annotations become more comprehensive, however, this user-driven approach quickly becomes inefficient for handling the vast amounts of video data available. In order to identify, for example, the types of shots that are encountered, or the relationships between components of the A/V stream, multiple (semi-)automatic analysis methods, each one fulfilling a distinct functionality, should be developed. Figure 1 is a generic depiction of this layered analysis approach, where a number of processing modules can coexist and operate on an input stream, either independently or in tandem. A variety of methods has been proposed in the literature for automatic extraction of the desired semantic information from a video sequence.

*Corresponding author.

†A. Murat Tekalp is currently with Koç University, Rumelifeneri Yolu, Sarıyer, Istanbul, Turkey.

Fig. 1 Possible processing blocks for semantic video analysis. Each block can function independently, or it can utilize the output of other modules; these potential dependencies are represented by the dashed arrows. The auxiliary input supplied to the system corresponds to such *a priori* information as user-specified domain models. The methods introduced in this paper are related specifically to the third processing module, “*Classify Individual Shots*.”

The majority of these techniques are *domain specific*; i.e., they are designed to operate on certain types of video data, and rely on *a priori* rules or models that dictate how these data should be interpreted. Recently there has been strong interest in the use of probabilistic methods—in particular, hidden Markov models (HMMs) and Bayesian belief networks (BBNs)—for video analysis. HMMs are useful tools for representing content domains where the types, as well as the temporal sequence, of shots are strictly constrained. The adopted model imposes these constraints by restricting the order and transition probabilities of the defined states. HMM-based methods have been employed in a variety of content-based video processing applications, including detection of scene transitions and simple camera movements,⁶ labeling individual shots in simple dialog scenes,⁷ and simultaneous temporal segmentation and frame labeling of broadcast news.⁸ The primary difficulty associated with the use of HMMs in semantic video analysis is the construction of the domain model itself, which requires extensive user input and supervision. Unless the targeted application involves a limited number of distinct camera setups that are repeated in a particular pattern, it is difficult to capture the temporal structure of the corresponding video process, or to obtain the accurate transition parameters upon training. Selection of the appropriate feature set for HMM applications introduces another source of complexity. In order to define a discrete symbol alphabet for observation sequences, most methods in the literature rely on *tokens* that are extracted from the original features; these tokens—rather than the features themselves—are then used by the model to determine the most probable sequence. Token computation introduces an intermediate processing and classification step,

which may itself involve such challenging tasks as face detection;⁷ whereas the use of continuous feature values requires modeling of the probability density function (pdf) for the observation vectors, and estimation of the pdf parameters. BBNs, on the other hand, provide a convenient graph structure for representation of causal relationships between multiple variables, allowing integration of prior knowledge from experts with evidence garnered from observations to infer the most probable outcome.⁹ While belief networks have been extensively employed in such areas as diagnosis and robotics, their application to semantic video processing has been limited to small-scale shot classification¹⁰ or modeling object relationships within a given shot.¹¹ This is mainly a consequence of the complexity of the video process in general, which renders identification and modeling of all causal dependences between the observed features, and assignment of prior and conditional probabilities, a considerable undertaking. The computational complexity of training and evidence propagation increases with the number of nodes in the network, which limits the scope of possible applications. Direct use of continuous-valued features is also difficult, which implies that a significant amount of processing and classification is required before the network can even be utilized.

In order to alleviate various shortcomings associated with existing analysis methods, we introduce in this paper a fuzzy framework for unsupervised classification of the shots within a video sequence. A high-level block diagram representation of the proposed method (which provides the functionality of the third processing module, “*Classify Individual Shots*,” in Fig. 1), is depicted in Fig. 2. During the *training* stage, a model for video data of interest is constructed using a relevant set of input samples and fuzzy clustering methods. Based on the size and scope of the training set, the resulting model—which comprises the number of distinct classes (e.g., the number of disparate shot types) and a corresponding set of labeled samples or prototypes that exemplify these classes—may apply to a single video sequence, a type of program, or an entire video content domain. The labeled training data are subsequently utilized to assign new input (from the same sequence, type of program, or domain) to the available classes by a fuzzy minimum-distance classifier. One of the main advantages of the proposed approach over existing methods is that it constructs a model of the input video automatically from training data, thereby eliminating the need for extensive user input and supervision. This is achieved through efficient use of clustering and cluster validation methods, whose application area have so far been limited to generation of compact, navigable summaries of video sequences.^{12–16} Further, we define a feature set that is based on common editing conventions, which allows us to capture and interpret the associations between neighboring shots in terms of generic low-level frame attributes. Domain-specific shot types and recurring temporal patterns are therefore uncovered by the computed feature values and clustering methods. If desired, additional information about a particular domain can be introduced to further enhance the final classification results.

Verbal description of the spatio-temporal properties of a video domain or its components is straightforward for viewers. However, translating these observations into quan-

Fig. 2 General block diagram of the proposed video analysis system.

titative rules for classification often presents a challenge. This lack of strict (quantitative) criteria implies that adoption of crisp classification methods, which associate each shot with a single class only, may lead to unnatural and erroneous conclusions. We avoid these potential pitfalls by describing the class affiliations of each shot using fuzzy labels, thereby providing to users a more detailed delineation of the given video domain. Furthermore, the attained membership values can be mapped to verbal descriptions, by means of a fuzzy linguistic variable, in order to approximate a qualitative interpretation of the classification results.

Overall, the classification system presents an intuitive, low-complexity method for processing and managing large amounts of video data. The resulting shot labels can be utilized directly, or in conjunction with a summarization algorithm, to automatically construct complete, browsable semantic descriptions and/or abstracts of video sequences. In particular, the algorithm can be deployed as an analysis tool for generating components of an MPEG-7-compliant description. Basic semantic elements of various MPEG-7 MDSs, such as the SummaryThemeList DS (which provides succinct textual themes or events to describe A/V content summaries), StructuredAnnotation datatype (which defines seven basic categories for textual content annotation), and PointOfView datatype (which describes the relative importance of an A/V segment given a specific viewpoint) can be instantiated for an input video sequence using the output of the proposed classification methods.

The rest of the paper is organized as follows. We introduce the set of editing-inspired features for shot content and relationship description in Sec. 2. Details of the clustering and cluster validation methods employed in the training stage, and additional editing-based information that can be used to improve the reference domain model, are given in Sec. 3. Section 4 outlines the properties of minimum-distance classifiers, and presents extensions that enable an end-to-end fuzzy representation. Experimental results that

demonstrate the performance of the algorithm are presented in Sec. 5.

2 Features for Shot Characterization

A primary goal of semantic video analysis is to reveal the associations and interactions between the shots in a video stream. Even simple juxtaposition prompts viewers to interrelate shots; such interactions are further strengthened through *associative cues*. Associative cues can be classified as *syntactic*, i.e., those involved purely with the spatial (motion-related) and tonal (color- and brightness-related) properties of a shot; or *semantic*, which are based primarily on semantic meaning and draw from the displayed environment and the personal experiences of viewers.¹⁷ The presentation order of the shots also plays a significant role in how a video stream is perceived and interpreted.¹⁸ Automatic analysis and identification of semantic associative cues is an unrealistic goal, since it requires quantitative modeling of numerous intangible concepts. It is, however, possible to computationally ascertain the shot relations suggested by syntactic cues, which can be approximated in terms of low-level image characteristics. This approach provides a simple yet general understanding of the semantic structure of a video sequence.¹⁹ Modeling and extraction of the syntactic associative cues between neighboring shots requires a detailed study of the *editing* process. The narrative structure of a film or TV program is largely established during editing; hence an understanding of widely employed editing “rules” or conventions will enable us to determine the proper set of low-level image features that should be extracted from a video sequence.²⁰ Variations in the editing practices applied in different video domains (e.g., a soap opera versus a live sports broadcast) lead to distinctive traits in the observed feature values, which can in turn be used to characterize each domain.

In the following sections we present several editing-related properties and corresponding low-level features that we use to represent, and capture the neighborhood relationships (i.e., associative cues) between, the elements (i.e., the constituent shots) of a video sequence. It is assumed that the locations of the shot boundaries in an input video sequence are made available to the classification system, having been obtained either through a temporal segmentation algorithm such as Ref. 21, or delivered as part of the video stream itself, e.g., with MPEG-7-encoded data.

2.1 Tonal Coherence

Tonal coherence, i.e., color-based similarity, between neighboring shots creates semantic association through a shared setting. We measure tonal coherence $C_{p,q}$ between two shots p and q as the $L1$ norm between their *alpha-trimmed average histograms* H_α^p and H_α^q . $H_\alpha^{(\cdot)}$ is obtained by sorting the array of frame histogram values for every bin in ascending order, and averaging only the central members of the ordered array.²² The trimmed mean operator²³ reduces the contribution of aberrant data points to the estimate by discarding an equal number of samples at each end of the sorted series. In practice, the effects of undesired luminance and chrominance variations within a shot (sudden flashes, occlusion of the camera, etc.) can be eliminated by modifying the trimming parameter α , $0 \leq \alpha \leq 0.5$, which controls the number of data points excluded from the average computation. Note that when $\alpha = 0$, H_α is the average histogram, while $\alpha = 0.5$ corresponds to the median histogram. Specific types of α -trimmed average histograms have been included among the MPEG-7 visual descriptors as the group-of-frames (GoF) color descriptor.²

2.2 Shot Motion Activity

Similarities in the amount of activity observed in temporally proximate shots often imply that the shots are semantically related, or that they belong to the same narrative unit. Motion activity within a shot can be calculated in a variety of ways. Due to its computational simplicity, we employ the average frame difference value $\mathcal{V}$ as the activity feature for a shot s :²⁴

$$\mathcal{V}_s = \frac{1}{|s|} \sum_{s_k, s_{k+1} \in s} \frac{\sum_{i,j=1}^{M,N} \mathcal{D}_{(s_k, s_{k+1})}(i,j)}{M \cdot N}, \quad (1)$$

where M and N denote the height and width, respectively, of a video frame in pixels. Motion-related variation over a frame pair (s_k, s_{k+1}) in s , $\mathcal{D}_{(s_k, s_{k+1})}$, is computed using the luminance value $\mathcal{P}(i,j)$ at each pixel location:

$$\mathcal{D}_{(s_k, s_{k+1})}(i,j) = \begin{cases} 1, & |\mathcal{P}_{s_{k+1}}(i,j) - \mathcal{P}_{s_k}(i,j)| > T \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where T is a predefined tolerance value. $\mathcal{V}$ effectively corresponds to the average percentage of pixels that change significantly per frame pair within s .

2.3 Shot Duration and Pacing

The duration of a shot influences the way it relates to shots before and after it, and helps define the rhythm in the scene. During editing, the temporal length of a shot is influenced by several factors, including the amount and clarity of information to be conveyed in the shot; picture quality (contrast, lighting level, etc.); and the level of action or variation throughout the shot.²⁵ Analysis of shot durations within a sequence can thus provide information about the content of individual shots, and also help distinguish coherent shot groupings.²⁶ Further statistical analysis can also yield valuable clues for film genre detection and style characterization.^{19,27,28}

For our application, we express the duration dur_s of a shot s in minutes; i.e.,

$$\text{dur}_s = \frac{\text{Number of frames in } s}{60 \cdot \text{frame rate (fps)}}. \quad (3)$$

Additionally, the ratio of the duration of s to those of its neighboring shots, i.e.,

$$\frac{\text{dur}_s}{\text{dur}_{s+k}}, k \geq 1$$

is also considered. It is expected that these ratios will both remain close to 1 for shots that belong to the same scene, while at the boundary of a scene more pronounced fluctuations will be observed.

2.4 Layout Similarity

When watching a program, the viewer retains a recollection of the spatial relationships between the objects observed across a series of shots;²⁵ the editor must therefore ensure that such established relationships (e.g., locations of people and camera angle) are accurately preserved. This is especially true for *continuity sequences*, such as dialog scenes, which feature several persistent setups that the camera switches between. Comparison of the compositions or layouts of neighboring shots can pinpoint these recurring shot types.²⁹

The *layout similarity* of two shots p and q (where p precedes q) is computed by comparing the “end” of p , p_e , to the “beginning” of q , q_b . The fundamental assumption here is that if p and q both have the same (or a similar) camera setup, their concatenation would yield a seamlessly integrated sequence of frames without a strongly discernible visual discontinuity. p_e and q_b may simply correspond to the first and last frames of the shots, respectively. Alternatively, composite frames may be generated from multiple frames at either end of a shot, to reduce the potential effects of camera and object motion.

Once the frames p_e and q_b are obtained, they are compared on a region-by-region basis, using the nine blocks shown in Fig. 3. The selection of these regions is based on the *Golden Section* spatial composition rule,²⁵ which suggests dividing up the screen in a 3:5 proportion (both horizontally and vertically), and positioning the main subjects on the intersections of these dividing lines, for an appealing visual arrangement. The proposed set of regions increases

Fig. 3 (a) The frame regions used for layout similarity comparison are shown overlaid on sample frames from the MPEG-7 test sequence *Camilo é Filho*, a sitcom. (b) The weights assigned to each of the nine regions are selected based on the location and size of each region.

the likelihood that all regions will contain valuable content information. Every region r is then assigned a weight ω_r based on its size and location in the frames, as shown in Fig. 3(b). Corresponding regions in the frame pair (p_e, q_b) are compared using the second-order statistical properties³⁰ of their chrominance characteristics:

$$\mathcal{L}_r(p, q) = \frac{\left[\frac{\sigma_{r(p_e)}^2 + \sigma_{r(q_b)}^2}{2} + \left(\frac{\mu_{r(p_e)} - \mu_{r(q_b)}}{2} \right)^2 \right]^2}{\sigma_{r(p_e)}^2 \cdot \sigma_{r(q_b)}^2},$$

$$r = 1, \dots, 9. \quad (4)$$

Here $r(\cdot)$ denotes the r th region in a frame, and $\mu_{r(\cdot)}$ and $\sigma_{r(\cdot)}^2$ are the mean and variance calculated over $r(\cdot)$, respectively. $\mathcal{L}$ has significant advantages over simple pixel-based comparison, in that it is less sensitive to the effects of object displacement. Minute variations in object locations that do not disturb the viewer are also tolerated by the similarity measure $\mathcal{L}$. Finally the feature values for each of the regions are combined in a weighted sum using $\omega_i, i = 1, \dots, 9$, to produce the layout similarity for the shots p and q :

$$\mathcal{L}(p, q) = \sum_{i=1}^9 \omega_i \mathcal{L}_i. \quad (5)$$

The set of features that we have defined mostly consider differences between neighboring shots; the main motivation for this approach is to render the attained representations independent of individual input sequences. These features can be computed over a user-defined temporal neighbor-

hood; i.e., a shot s is compared to each of its $l \geq 1$ successors, which yields, for every difference feature, an l -dimensional vector. These feature vectors are subsequently concatenated to form a single pattern vector for every shot, whose dimension $L \geq l$ is dependent on the number of features used and the selected neighborhood l . Note that not all features may be useful for every domain of interest; for example, information about shot durations is a significant cue for movies and action dramas, but not for talk shows. The set of features that are appropriate for a particular video domain can be defined by a user, or automatically determined through feature analysis/selection methods^{31,32} during the training phase.

The majority of the features in Secs. 2.1–2.4 can be derived directly from corresponding MPEG-7 visual descriptors, such as the GoF color histograms that are used to calculate tonal coherence. Alternative features that offer identical or similar functionality to those above can also be obtained from their MPEG-7 counterparts. For example, shot activity can be expressed in terms of the MotionActivity descriptor, which is designed to capture the level of activity within a segment. For layout similarity, visual descriptors such as the GridLayout D (which allows an image or frame to be split into rectangular regions that can be described separately) and RegionLocator D (which enables specification of polygon-shaped areas within an image) can be used to define the set of regions depicted in Fig. 3(b). The close compatibility of the proposed feature set with MPEG-7 visual descriptors provides flexibility for different implementations, and ensures that the amount of computation for feature extraction will be significantly reduced when these descriptors are accessible as part of an MPEG-7 stream.

3 Training and Domain Model Construction

The training stage of the algorithm requires that a comprehensive set of classes, which reflects the properties of the different shot types in the video data of interest, be identified, along with labeled observation vectors or prototypes for each class. Ideally each defined shot category should correspond to a compact cluster that is sufficiently separated from other clusters in the feature space, to minimize redundancy in the model and to ensure that new input data can be accurately classified. The simplest way to achieve these goals is to have an expert first select the appropriate number of classes that best characterizes the video data (e.g., a type of program), and then manually assign each element of a training set X to the proper class. This approach, however, demands a significant amount of manual labor for each particular sequence or domain of interest. Furthermore, such expert-driven model declaration is inevitably subjective, which increases the likelihood that inconsistent criteria will be employed during modeling of different video content. To sidestep these issues, we employ fuzzy clustering and cluster validation methods to automatically construct a reference model for the video data from the training set. The fuzzy clustering method offers a more accurate and detailed perspective about the data set by establishing to what degree a sample belongs to each of the available classes.³³ Given multiple partitions of a data set (each of which may contain a different number of clusters), cluster validation techniques are used to rank these partitions according to various criteria and choose the optimal one.³⁴

Selection of the training set X is of paramount importance, since it directly impacts the quality of the resulting model and subsequent classification performance. The members of X should accurately reflect the common properties of the video data to be modeled, so that the clusters and their members can be used as an unbiased model for the classification stage. The properties of the training set are also impacted by the scope of the targeted application. For a small-scale task, where the goal is to label the shots of an arbitrary video sequence without domain knowledge, part(s) of the sequence can be used as the training data to classify the remaining portion. In contrast, a larger training set is required if the aim is to obtain an accurate model of a video domain.

3.1 Domain Modeling Through Cluster Validation Methods

In order to construct the reference model of the video data from the available training set, first the fuzzy c -means (FCM) algorithm³⁵ is used to successively partition X for a set of values of c , with $c = 2, \dots, N, N \leq n = |X|$. Each resulting partition of the data set is compactly expressed as a $(c \times n)$ partition matrix $U = [u_{ij}]$, where $u_{ij} \in [0, 1]$ denotes the (fuzzy) membership of sample j in class i , and satisfies

$$\sum_{i=1}^c u_{ij} = 1, \forall j. \quad (6)$$

The membership values determine how strongly a sample point belongs in, or resembles, each of the available

classes. Given the unlabeled data set X , where each data point is represented by a t -dimensional feature vector $\mathbf{x}_j$, the FCM algorithm minimizes the objective function

$$J_m = \sum_{j=1}^n \sum_{i=1}^c (u_{ij})^m \|\mathbf{x}_j - \mathbf{v}_i\|_A^2. \quad (7)$$

Here $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c)$ is a vector of cluster centers, with

$$\mathbf{v}_i = \frac{\sum_{j=1}^n (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^n (u_{ij})^m};$$

$\|\mathbf{x}\|_A = \sqrt{\mathbf{x}^T A \mathbf{x}}$ denotes an inner product norm with A a $t \times t$ positive definite matrix; and m is the fuzzy weighting exponent that controls the level of membership sharing between clusters. (In our experiments the value of m has been uniformly set to 2.) J_m is minimized through an iterative process.

The *validity* of each resulting partition is then computed in terms of the Xie–Beni index, v_{XB} ,³⁶ which is defined (for $m = 2$) as

$$v_{XB}(U, \mathbf{V}; X) = \frac{\sum_{i=1}^c \sum_{j=1}^n u_{ij}^2 \|\mathbf{x}_j - \mathbf{v}_i\|^2}{n(\min_{i \neq l} \{\|\mathbf{v}_i - \mathbf{v}_l\|^2\})} \quad (8)$$

$$= \left[\frac{\sigma}{n} \right]_{\text{sep}(\mathbf{V})} \quad (9)$$

with

$$\sigma(U, \mathbf{V}; X) = \sum_{i=1}^c \left(\sum_{j=1}^n u_{ij}^2 \|\mathbf{x}_j - \mathbf{v}_i\|^2 \right); \quad (10)$$

$$\text{sep}(\mathbf{V}) = \min_{i \neq l} \{\|\mathbf{v}_i - \mathbf{v}_l\|^2\}. \quad (11)$$

An extended form of v_{XB} can be obtained when u_{ij}^2 is replaced by u_{ij}^m . A small σ indicates a good $(U, \mathbf{V})$ pair, since u_{ij} attains a high value when the distance $\|\mathbf{x}_j - \mathbf{v}_i\|$ between sample j and the centroid of cluster i is small. On the other hand, when the cluster centroids are well separated, $\text{sep}(\mathbf{V})$ will be large. Thus, given two different partitions U_1 and U_2 for the same data set, when $v_{XB}(U_1, \mathbf{V}_1; X) < v_{XB}(U_2, \mathbf{V}_2; X)$ due to either one (or both) of the above reasons, U_1 is taken to be a better partition than U_2 . The partition U^* of the training set X that minimizes v_{XB} , and the corresponding number of classes c^* , are finally adopted as the reference model of the given video sequence or domain.

Members of the resulting set of clusters can be further analyzed to pick out, for each class, a core set of labeled observations or prototypes. Within a class i , the elements whose memberships in i exceed a threshold $\rho \in [0, 1]$ are assumed to more faithfully represent the class properties. Using this reduced set to label new input samples in later processing stages is likely to improve classification accu-

racy. Core members for each cluster i can be directly selected by a thresholding operation on u_{ij} , with the value of ρ exceeding $1/c^*$. Additionally, multiple clusters may be merged or declared as a single class based on user input, to better reflect the semantic properties of the video sequence or domain of interest. This “intervention” mechanism allows expert feedback to be incorporated into the training process, even though such supervision is not required for the algorithm to function.

The set of labeled training samples are then used in a minimum-distance classification scheme to assign new input data to the identified classes. In Sec. 4 we review the properties of minimum-distance classifiers, and focus in particular on fuzzy extensions of nearest-prototype and nearest-neighbor machines.

3.2 Model Enhancement Through Domain-Dependent Editing Cues

In addition to the low-level visual features presented in Secs. 2.1–2.4, other editing-related information can be considered to augment and improve the acquired domain models. While these cues may not be conveniently accommodated within the clustering framework, they can be introduced as a set of rules that exploit specific properties of the input data. Such auxiliary information is often relevant for a particular video domain, and is therefore appropriate for enhancing and verifying the classification results within that domain only.

3.2.1 Shot transition methods

Different types of shot transition methods carry certain semantic implications for video content with narrative structure (e.g., movies, TV dramas, sitcoms, etc.). In this section we consider three of the most commonly used shot transition methods: cuts, fades, and dissolves. (While wipe is another popular shot transition method, it is used less frequently than those reviewed here. Furthermore, in most cases a wipe can be replaced by a dissolve without a loss of functionality or lapse in continuity.)

- The *cut* is generally used to connect continuous activity in the present tense, or to relate parallel actions that occur at the same time. The cut may also be used to connect shots at different periods of time, where the time lapse is generally on the order of hours. When used in such a context, the connected shots exhibit strong contrast in tone or activity, to hint that the continuity implied by the cut is interrupted.
- *Fades* have the effect of establishing the boundaries of individual scenes. A fade-out in a film generally corresponds to the endpoint of the respective scene, while the fade-in signals the start of a new scene. The fade introduces a pause in the flow of action, and notifies the viewer that a significant change in content is about to occur.
- A *dissolve* may be used to convey the passage of time, offer a condensed summary of an incident, or to bridge events occurring at separate times and places. If a series of consecutive dissolves are observed, this is taken as passing/contraction of time or a flashback,

and the same sequence is assumed to continue.^{26,37} A single dissolve between straight cuts, on the other hand, is an indication of a new scene.

Based on the above observations, it is possible to define a set of simple rules that amend the annotation of the shots that are associated with fades and dissolves,²⁰ such as labeling the shots that immediately follow a fade-in as the start of a new scene. Rules involving edit effects can be defined for other content domains also (e.g., within a sports program, dissolves and similar transition effects often correspond to the start/end of replay sequences).

3.2.2 Establishing shots

An establishing shot is (typically) a long shot that introduces the location where an action takes place;³⁸ thus it carries significant spatial content information. It also signals the starting point of a group of shots that take place within the same spatio-temporal window and are closely associated in narrative content. Furthermore, the same establishing shot, or similar ones of the same location, may be used repeatedly throughout the content. Establishing shots generally possess the following visual attributes:

1. they exhibit low tonal variation;
2. there is limited camera movement and the level of activity is low;
3. they are highly different (in color and motion content) from shots that precede and follow them; and
4. certain edit effects (fades and dissolves) are employed in conjunction with them.

The first three properties listed above are addressed by the low-level features introduced in the earlier sections, while the latter trait can be captured through an analysis of the observed shot transition methods.

4 Minimum-Distance Classification

Nearest-prototype (1-NP) and nearest-neighbor (1-NN) classifiers are formulated in terms of the distances between observations to be classified and the paradigms of the available classes in the feature space.³⁵ In the 1-NP case, these paradigms are the prototypical vectors $\alpha_i, i = 1, \dots, c$, while for a 1-NN classifier, they are a set of labeled observations. The decision rule for a nearest-prototype classifier is given by³¹

$$\text{Decide } \mathbf{x} \in i \Leftrightarrow d(\mathbf{x}, \alpha_i) = \min_{1 \leq q \leq c} \{d(\mathbf{x}, \alpha_q)\}, \quad (12)$$

where $\mathbf{x}$ is the feature vector of the input sample, and d denotes an appropriate distance function. If each class is represented by multiple prototypes, the distance between a sample j and class i can be taken as the minimum of the distances from j to each of the prototypes of i . Alternatively, j can be assigned to the class that is observed most frequently among the K prototypes closest to it.³⁵ For the nearest-neighbor classifier, the decision rule of Eq. (12) becomes³⁹

$$\text{Decide } \mathbf{x} \in i \Leftrightarrow d(\mathbf{x}, \mathbf{x}_i) = \min_{1 \leq q \leq n} \{d(\mathbf{x}, \mathbf{x}_q)\}, \quad (13)$$

where $\mathbf{x}_i$ denotes the feature vector of a labeled observation assigned to class i , and n is the number of such observations available. The 1-NN classifier can likewise be extended to render its decision based on the K nearest neighbors of a sample; an input pattern is assigned to the class that is encountered the most among these K nearest labeled neighbors.

The classification rules in Eqs. (12) and (13) define *crisp* classifiers; i.e., each input sample is affiliated with a single class only, by means of a binary membership function $u_{ij} \in \{0,1\}$. Furthermore, the crisp decision criteria are based solely on the distance of a sample to the class paradigms, and do not take into account any continuous membership values that may be available. In order to implement these classifiers it is required that the fuzzy partition obtained during the training stage first be converted to a crisp partition. Incorporating fuzzy set theory into minimum-distance classifier design has obvious advantages for pattern recognition applications, where an observation is typically similar to more than a single class. Fuzzy techniques often improve classification accuracy, since they avoid unrealistic absolute class assignments and, through the continuous membership values, provide a measure of confidence in the classification. Various nearest-prototype and nearest-neighbor classifiers that return fuzzy decisions based on fuzzy labels have been proposed in the literature;^{33,40,41} in the following paragraphs we review those presented by Keller *et al.* in Ref. 33.

For the fuzzy nearest-prototype classifier, the membership of an input pattern in class i is computed by

$$u_i(\mathbf{x}) = \frac{1}{\left[\sum_{l=1}^c \left(\frac{d(\mathbf{x}, \boldsymbol{\alpha}_l)}{d(\mathbf{x}, \boldsymbol{\alpha}_i)} \right)^{2/(m-1)} \right]}. \quad (14)$$

Membership assignments are based solely on the distance of an input sample to the prototype vectors, since each prototype is associated only with the class that it represents.³³ Equation (14) can easily be extended to handle multiple prototypes by modifying the employed distance function. The fuzzy K nearest-neighbor classifier, on the other hand, determines the class affiliations of an input sample as

$$u_i(\mathbf{x}) = \frac{\sum_{l=1}^K u_{il} \cdot \hat{d}(\mathbf{x}, \mathbf{x}_l)}{\sum_{l=1}^K \hat{d}(\mathbf{x}, \mathbf{x}_l)}, \quad (15)$$

where

$$\hat{d}(\mathbf{x}, \mathbf{x}_l) = \frac{1}{d(\mathbf{x}, \mathbf{x}_l)^{2/(m-1)}}, \quad (16)$$

where $u_{il}, l=1, \dots, K$, denote the class memberships of the K labeled observations closest to the input sample, which

Table 1 Properties of the test sequences used in the experiments.

Sequence	Domain type	Number of shots
CF	Sitcom	327
PP	Sitcom	135
LS	TV movie	101
SG	Game show	135
S	Sitcom	1013
MLB	Baseball (MLB)	848

are weighted by their distances to $\mathbf{x}$. The denominator in Eq. (15) is a normalization coefficient to satisfy the constraint $\sum_{i=1}^c u_i(\mathbf{x}) = 1$.

5 Experimental Results

The proposed methods were tested on several sequences selected from the MPEG-7 content set (Table 1)⁴²: *Camilo é Filho* (CF) and *Pepa y Pepe* (PP), both sitcoms; *La Sombra de un Ciprés es Alagarda* (LS), a TV movie; and *Saber y Ganar* (SG), a TV game show. In addition to these sequences, six different episodes of the syndicated sitcom *Seinfeld* (S) and seven 20 min clips of major league baseball games (MLB) were captured from broadcast TV. The shot boundaries for the sequences were first identified with the automatic shot boundary detection tools described in Ref. 21 and subsequently refined by a human operator. The low-level visual features of Secs. 2.1–2.4 were then computed for a shot neighborhood of $l=3$, yielding a ten-element feature vector for every shot. The training sets were partitioned using the fuzzy c -means algorithm, with $c=2, \dots, 20$; the partition that minimized the Xie–Beni index, and the corresponding number of classes c^* , were finally adopted as the representative model for the video content. New input data were assigned to these c^* categories using both crisp and fuzzy K -NN classifiers, with $K=3$. For the fuzzy nearest-neighbor classifier, each shot was assigned to the class of maximum membership if this membership value exceeded 0.5; otherwise, the shot was marked as *unassigned*. The resulting shot labels were finally inspected by a human operator to determine the number of correct and erroneous classifications.

In our experiments we have considered three distinct applications: (i) Single-sequence classification, where a portion of the input sequence is used as the training set to first generate the reference model for the input program. The rest of the shots are subsequently labeled using this model and fuzzy nearest-neighbor classification methods. (ii) Single-program classification, where video data from different episodes of a particular program are labeled, based on training data that also belong to the same program. Note that this approach is a direct extension of single-sequence classification. (iii) Single-domain classification, where the training set and the unlabeled data do not necessarily come from the same program, but rather belong to the same video domain.

The results of single-sequence classification experiments are shown in Tables 2 and 3 for crisp and fuzzy nearest-neighbor classifiers, respectively. For each sequence, roughly half of the shots were randomly selected and uti-

Table 2 Results of the single-sequence classification experiments using a crisp K -NN classifier with $K=3$. Here $|X|$ is the number of elements (i.e., shots) in the training set; c^* is the optimal number of classes determined by the Xie–Beni index; and $|Y|$ is the number of unlabeled input samples. C and F denote the number of correct and false classifications, respectively.

Sequence	$ X $	c^*	$ Y $	C	F
CF	150	8	177	131	46
LS	50	12	51	37	14
SG	60	5	75	68	7
PP	65	6	70	54	16

lized during the training stage. It is observed that the number of misclassifications are significantly reduced with the fuzzy approach: 19.57% of the assignments made by the crisp classifier are incorrect, compared to only 7.74% for the fuzzy machine. This is a direct consequence of the continuous membership values, which serve as “confidence measures” that help determine whether a label assignment is justified for an input sample. The shots that fall into the *unassigned* category can be further analyzed to determine if they correspond to distinct clusters in the feature set that constitute a new class, which was undetected in the training stage.

The number of classes established for the training set through cluster validation methods accurately captures the properties of each individual domain or program type. The game show comprises static, often-repeated camera setups; this fact is reflected by the small c^* value that is automatically determined by the algorithm. The movie sequence, on the other hand, features a large number of locations and different types of shots; consequently, the program model contains significantly more classes. For the sitcom sequence CF, a total of eight classes were identified, representatives from which are depicted in Fig. 4. Since the employed feature set highlights the neighborhood relationships between successive shots, the classes in the model pinpoint various types of shot “patterns” that are encountered in the sequence, which we use to label the obtained classes. Some of the classes actually correspond to the same type of pattern; if desired, these clusters can be merged (through explicit user input). Rather than merging them, we have cho-

Table 3 Results of the single-sequence classification experiments using a fuzzy K -NN classifier with $K=3$. Here $|X|$ is the number of elements (i.e., shots) in the training set; c^* is the optimal number of classes determined by the Xie–Beni index; and $|Y|$ is the number of unlabeled input samples. C and F denote the number of correct and false classifications, respectively, and U is the number of unassigned samples.

Sequence	$ X $	c^*	$ Y $	C	F	U
CF	150	8	177	123	14	40
LS	50	12	51	33	6	12
SG	60	5	75	65	3	7
PP	65	6	70	54	6	10

sen to assign the members of both these classes the same descriptive label.

Single-program classification experiments were carried out using the test data obtained from the six episodes of the syndicated sitcom S. Table 4 summarizes the results of the experiments, using both crisp and fuzzy classifiers. The fuzzy approach again reduces the number of misclassifications that are observed when a crisp classifier is utilized. It is interesting to note that the classification performance (i.e., the number of erroneous classifications) for the single-program case is slightly better than that for single-sequence classification. This indeed is not surprising, since, along with common camera setups and shot types, all S episodes (which are all from the same season) feature a similar set of locations (e.g., Jerry’s apartment; Monk’s; Susan’s apartment; etc.). Hence the training set (which comprises two of the six episodes), and the resulting domain model with 14 identified shot categories, characterizes the entire data set quite accurately.

In order to test single-domain classification, shots from the sitcom PP were labeled using the domain model constructed with data from CF. As seen in Table 5, the new input samples are labeled with high accuracy, which implies that the proposed feature set is able to characterize a particular program domain. It is worthwhile to observe that sequence PP is less “complex” compared to sequence CF; i.e., it contains fewer shot types, which ensures that the available domain model covers all instances observed in PP. Figure 5 illustrates three sample shots from PP that are assigned the correct class labels.

Single-domain classification experiments were also carried out using test data obtained from MLB game broadcasts. The sequences (approximately 112 min of baseball footage, after commercials were eliminated) were collected from four different games on two different cable TV channels (ESPN and TBS), and about a quarter of the 848 extracted shots were randomly selected to constitute the training set. Baseball provides a good content type for the proposed approach, since (i) it is more structured than other types of sports (e.g., soccer), and (ii) game broadcasts often feature a limited number of camera angles and shot types that remain constant even across different TV channels. The results of crisp and fuzzy nearest-neighbor classification experiments are given in Table 6. The fuzzy classification approach significantly reduces the misclassifications that are present when a crisp classifier is employed. Note that the number of shot types identified during the training stage (namely, 18) is quite large. A significant portion of these categories corresponds to shots that are unrelated to the game itself (such as spectators, commentators, etc.) and can be eliminated through user input after the initial classification stage. This refinement step yields only those shot types that are interesting to the user (e.g., pitches, hits, home runs, etc), and allows the input shots to be “filtered” during the classification process.

The reference models that are identified during the training stage often vary with l , the size of the shot neighborhood that is used during feature computation. This parameter is influenced by the properties of the video domain under consideration: It is advantageous to select a larger neighborhood (e.g., $l=3$ or $l=4$) for such domains as sitcoms, talk shows, and game shows, where the same shots

Fig. 4 Representative shots from the eight identified classes in the model for *Camilo é Filho*. Immediate neighbors of the representative shots and descriptive pattern labels are shown for each class.

Table 4 Results of the crisp and fuzzy classification experiments for the *Seinfeld* episodes. The domain model was first constructed using two of the available six episodes (320 of the available 1013 shots). Here $|X|$ is the number of elements (i.e., shots) in the training set; $|Y|$ denotes the number of unlabeled input samples from S ; and c^* is the number of categories in the domain model, which was determined through cluster validity methods to be 14. C and F denote the number of correct and false classifications, respectively, and U is the number of unassigned samples for the fuzzy classifier. A K -NN classifier with $K=3$ was used in both crisp and fuzzy experiments.

Sequence	$ X $	c^*	$ Y $	C	F	U
S (crisp)	320	14	693	592	101	N/A
S (fuzzy)	320	14	693	587	38	68

Table 5 Results of the classification experiments for the sequence *Pepa y Pepe*. Each shot from the sequence was classified using the domain model obtained using a training set from *Camilo é Filho*, another sitcom. $|Y|$ denotes the number of unlabeled input samples from *Pepa y Pepe*, and c^* is the number of categories in the domain model. C and F denote the number of correct and false classifications, respectively, and U is the number of unassigned samples. A fuzzy K -NN classifier with $K=3$ was used in the experiment.

Sequence	c^*	$ Y $	C	F	U
PP	8	135	106	11	18

Fig. 5 Test shots from the sequence *Pepa y Pepe*, and the labels assigned to each one based on the domain model extracted from another sitcom, *Camilo é Filho*.

are repeated often. For more dynamic program types, such as dramas and films, however, l should be set to 1 or 2; otherwise the number of classes in the constructed model may become larger.

A subset of the available feature set is generally sufficient to characterize a particular domain. It is therefore important to accurately identify, for each domain or program type of interest, those measurements that will help discriminate between the clusters inherent in the data set. The variance value σ^2 is often a good indicator of a feature's significance. A small σ^2 implies that the feature remains more or less uniform throughout the data set; in this case, the feature can be ignored without a noticeable loss in performance. Table 7 illustrates the σ values for the features extracted from the sitcom sequence, CF. Compared to the rest of the features, the variance of the activity feature is very low, which implies that its contribution to cluster separation

in the feature space is limited. In addition to variance analysis, basic background information about individual program types can also be considered when choosing the appropriate feature set for a particular domain. For example, tonal coherence and layout similarity features are often sufficient for modeling static domains such as talk shows, while shot duration and activity features are more valuable for movies and TV dramas, where the duration of programs, as well as the number of locations and scenes, increase considerably.

6 Conclusions

A fuzzy classification method has been developed for unsupervised domain-dependent video sequence analysis and shot labeling. The proposed approach significantly reduces the amount of user input required to establish a reference model for the video data of interest, through the use of clustering and cluster validation methods. A fuzzy nearest-neighbor/prototype classifier is then used to assign unlabeled observations to the available classes. The fuzzy membership values associated with each sample provide a confidence measure for the class assignments, and allow erroneous classifications to be avoided when an input sample lacks strong affiliation with any of the categories. The resulting class labels identify the neighborhood relationships and recurring shot groups within a video sequence or domain. The algorithm offers an efficient tool for gener-

Table 6 Results of the crisp and fuzzy classification experiments for the *MLB* data. The domain model was first constructed using about a quarter of the available test set (i.e., $|X|=225$). $|Y|$ denotes the number of unlabeled input samples from S , and c^* is the number of categories in the domain model, which was determined through cluster validity methods to be 18. C and F denote the number of correct and false classifications, respectively, and U is the number of unassigned samples for the fuzzy classifier. A K -NN classifier with $K=3$ was used in both crisp and fuzzy experiments.

Sequence	$ X $	c^*	$ Y $	C	F	U
MLB (crisp)	225	18	623	514	109	N/A
MLB (fuzzy)	225	18	623	504	43	76

Table 7 Sample standard deviations of the (first-order) features computed for the test sequence *Camilo é Filho*.

	$\mathcal{C}$	$\mathcal{L}$	Duration	$\mathcal{V}$
σ	0.1117	1.5396	0.1067	0.0393

ating basic semantic video annotations, and in particular various components of MPEG-7-compliant content descriptions.

Acknowledgment

This work was supported by a National Science Foundation SIUCRC grant and a New York State Office for Science, Technology and Academic Research grant to the Center for Electronic Imaging Systems at the University of Rochester.

References

1. J. M. Martinez, "Overview of the MPEG-7 Standard (v. 5.0)," ISO/JTC1/SC29/WG11/N4031, Singapore (March 2001).
2. ISO/IEC, Final Committee Draft 15938-3, "Information Technology—Multimedia content description interface: Visual," ISO/JTC1/SC29/WG11/N4062, Singapore (March 2001).
3. ISO/IEC, Final Committee Draft 15938-5, "Information Technology—Multimedia content description interface: Multimedia Description Schemes," ISO/JTC1/SC29/WG11/N3966, Singapore (March 2001).
4. ISO/IEC, Final Committee Draft 15938-2, "Information Technology—Multimedia content description interface: Description Definition Language," ISO/JTC1/SC29/WG11/N4002, Singapore (March 2001).
5. W3C, "XML Schema Part 0: Primer, W3C Recommendation," <http://www.w3.org/TR/xmlschema-0/> (May 2001).
6. J. S. Boreczky and L. D. Wilcox, "A hidden Markov framework for video segmentation using audio and image features," in *Proc. ICASSP*, Vol. 6, pp. 3741–3744, Seattle (May 1998).
7. W. Wolf, "Hidden Markov model parsing of video programs," in *Proc. ICASSP*, Vol. 4, pp. 2609–2611, Munich (April 1997).
8. S. Eickeler and S. Muller, "Content-based video indexing of TV broadcast news using hidden Markov models," in *Proc. ICASSP*, Vol. 6, pp. 2999–3000, Phoenix (Mar. 1999).
9. F. V. Jensen, *Introduction to Bayesian Networks*, Springer, New York (1996).
10. N. Vasconcelos and A. Lippman, "Bayesian modeling of video editing and structure: semantic features for video summarization and browsing," in *Proc. IEEE ICIP*, Vol. 3, pp. 153–157, Chicago (Oct. 1998).
11. A. M. Ferman and A. M. Tekalp, "Probabilistic analysis and extraction of video content," in *Proc. IEEE ICIP*, Kobe, Japan (Oct. 1999).
12. D. Zhong, H. Zhang, and S.-F. Chang, "Clustering methods for video browsing and annotation," in *Proc. Storage and Retrieval for Image and Video Databases IV*, *Proc. SPIE* **2670**, 239–246 (1996).
13. M. M. Yeung and B. Liu, "Efficient matching and clustering of video shots," in *Proc. IEEE ICIP*, Vol. 1, pp. 338–341 Washington DC, (Oct. 1995).
14. M. M. Yeung and B. L. Yeo, "Time-constrained clustering for segmentation of video into story units," in *Proc. ICPR*, pp. 375–380, Vienna, Austria (August 1996).
15. M. M. Yeung, B.-L. Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," in *Multimedia Networking and Computing*, *Proc. SPIE* **2417**, 399–413 (Feb. 1995).
16. A. Hanjalic and H. J. Zhang, "An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis," *IEEE Trans. Circuits Syst. Video Technol.* **9**(8), 1280–1289 (1999).
17. J. P. Isenhour, "The effects of context and order in film editing," *AV Commun. Rev.* **23**(1), 69–80 (1975).
18. S. Worth, "Cognitive aspects of sequence in visual communication," *AV Commun. Rev.* **16**(2), 121–145 (1968).
19. B. Salt, *Film Style and Technology: History and Analysis*, 2nd ed., Starwood, London (1992).
20. A. M. Ferman and A. M. Tekalp, "Editing cues for content-based analysis and summarization of motion pictures," in *Storage and Retrieval for Image and Video Databases VI*, *Proc. SPIE* **3312**, 71–80 (Jan. 1998).
21. A. M. Ferman and A. M. Tekalp, "Efficient filtering and clustering methods for temporal video segmentation and visual summarization," *J. Visual Commun. Image Represent* **9**(4), 336–351 (Dec. 1998).
22. A. M. Ferman, A. M. Tekalp, and R. Mehrotra, "Robust color histogram descriptors for video segment retrieval and identification," *IEEE Trans. Image Process.* (submitted).
23. J. B. Bednar and T. L. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Trans. Acoust., Speech, Signal Process.* **ASSP-32**(1), 145–153 (1984).
24. H. Zhang, A. Kankanhalli, and S. Somaliar, "Automatic partitioning of full-motion video," *ACM/Springer Multimedia Sys. J.* **1**(1), 10–28 (1993).
25. G. Millerson, *The Technique of Television Production*, 12th ed., Focal (March 1990).
26. P. Aigrain, P. Joly, P. Lepain, and V. Longueville, "Representation-based user interfaces for the audiovisual library of the year 2000," in *Multimedia Computing and Networking*, *Proc. SPIE* **2417**, 35–45 (1995).
27. D. C. Coll and G. K. Choma, "Image activity characteristics in broadcast television," *IEEE Trans. Commun.* **27**, Oct., 1201–1206 (1976).
28. S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," in *Proc. ACM Multimedia'95* pp. 295–304, San Francisco, CA (Nov. 1995).
29. H. Aoki et al., "A shot classification method of selecting effective key frames for video browsing," in *Proc. ACM Multimedia '96*, Boston, MA (1996).
30. R. Kasturi and R. Jain, "Dynamic vision," in *Computer Vision: Principles*, pp. 469–480, R. Kasturi and R. Jain, Eds., IEEE Computer Society, Washington (1991).
31. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley, Reading, MA (1974).
32. K. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Kluwer Academic, Boston, MA (1998).
33. J. M. Keller, M. R. Gray, and J. A. Givens, Jr., "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Syst. Man Cybern.* **SMC-15**(4), 580–585 (1985).
34. N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.* **3**(3), 370–379 (1995).
35. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York (1981).
36. X. L. Xie and G. A. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.* **3**(8), 841–846 (1991).
37. S. Butler and A. P. Parkes, "Filmic time-space diagrams for video structure representation," *Signal Process. Image Commun.* **8**, 269–280 (1996).
38. B. F. Kavin, *How Movies Work*, Macmillan, New York (1987).
39. T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory* **IT-13**, 21–27 (1967).
40. L. I. Kuncheva and James C. Bezdek, "An integrated framework for generalized nearest prototype classifier design," *Int. J. Uncertainty, Fuzziness, Knowledge-Based Syst.* **6**(5), 437–457 (1998).
41. J. H. Han and Y. K. Kim, "A fuzzy K-NN algorithm using weights from the variance of membership values," in *Proc. IEEE Int. Conf. on Comp. Vision and Pattern Rec.*, pp. 394–399, Fort Collins (June 1999).
42. MPEG-7 Video Group, Licensing Agreement for the MPEG-7 content set, ISO/JTC1/SC29/WG11/N2466, Atlantic City (Oct. 1999).

A. Müfit Ferman received his BSc and MSc degrees in electronics and communication engineering from Istanbul Technical University, Turkey, in 1993 and 1995, and his MS and PhD degrees in electrical engineering from the University of Rochester, New York, in 1997 and 2001, respectively. From February 1994 to August 1995 he was a research and teaching assistant at the Department of Electronics and Communication Engineering at Istanbul Technical University. In September 1995 he joined the Department of Electrical and Computer Engineering at the University of Rochester as a PhD candidate and graduate assistant. Since August 2000 he has been with Sharp Laboratories of America, Inc. in Camas, WA. His research interests include content-based video indexing and retrieval, visual information management systems, data mining, and pattern recognition applications. Dr. Ferman is a member of the ACM, IEEE, and SPIE.

A. Murat Tekalp received his BS degrees in electrical engineering and mathematics from Bogazici University, Istanbul, Turkey, in 1980, with the highest honors, and his MS and PhD degrees in electrical, computer, and systems engineering from Rensselaer Polytechnic Institute (RPI), Troy, New York, in 1982 and 1984, respectively. From December 1984 to August 1987, he was a research scientist and then a senior research scientist at Eastman Kodak Company, Rochester, New York. He joined the Electrical and Computer Engineering Department at the University of Rochester, Rochester, New York, in September 1987, where he is currently a distinguished university professor. He is also affiliated with the Koç University, Istanbul, Turkey. His current research interests are in digital image

and video processing, including image/video restoration, video segmentation, object tracking, video coding, content-based video description, and secure media. Professor Tekalp is a senior member of IEEE. He received the NSF Research Initiation Award in 1988, and named distinguished lecturer by the IEEE Signal Processing Society in 1998. He has chaired the IEEE Signal Processing Society Technical Committee on Image and Multidimensional Signal Processing (Jan. 1996–Dec. 1997) and is a founding member of the Technical Committee on Multimedia Signal Processing. He has served as an associate editor for the *IEEE Trans. on Signal Processing* (1990–1992), *IEEE Trans. on Image Processing* (1994–1996), and the Kluwer journal *Multidimensional Systems and Signal Processing* (1994–1999). He was an area editor for the Academic Press journal *Graphical Models and Image Processing* (1995–1998). He was also on the editorial board of the Academic Press journal *Visual Communication and Image Representation* (1995–1999). He was appointed

as the Technical Program chair for the 1991 IEEE Signal Processing Society Workshop on Image and Multidimensional Signal Processing and the Special Sessions chair for the 1995 IEEE International Conference on Image Processing and Technical Program co-chair for IEEE ICASSP 2000 in Istanbul, Turkey. He is the founder and first chairman of the Rochester chapter of the IEEE Signal Processing Society. He was elected as the chair of the Rochester Section of IEEE in 1994–1995. At present, he is the editor-in-chief of the EURASIP journal on Image Communication published by Elsevier Science. He is also the general chair of the IEEE International Conference on Image Processing to be held in Rochester, NY in September 2002. He authored the Prentice Hall book *Digital Video Processing* (1995). Dr. Tekalp holds five US patents. His group contributed technology to the ISO/IEC MPEG-4 and MPEG-7 standards.