MPEG VBR Video Traffic Modeling and Classification Using Fuzzy Technique

Qilian Liang and Jerry M. Mendel

***Abstract*—In this paper, we present a new approach for MPEG variable bit rate (VBR) video modeling and classification using fuzzy techniques. We demonstrate that a type-2 fuzzy membership function, i.e., a Gaussian MF with uncertain variance, is most appropriate to model the log-value of I/P/B frame sizes in MPEG VBR video. The fuzzy c-means (FCM) method is used to obtain the mean and standard deviation (std) of I/P/B frame sizes when the frame category is unknown. We propose to use type-2 fuzzy logic classifiers (FLCs) to classify video traffic using compressed data. Five fuzzy classifiers and a Bayesian classifier are designed for video traffic classification, and the fuzzy classifiers are compared against the Bayesian classifier. Simulation results show that a type-2 fuzzy classifier in which the input is modeled as a type-2 fuzzy set and antecedent membership functions are modeled as type-2 fuzzy sets performs the best of the five classifiers when the testing video product is not included in the training products and a steepest descent algorithm is used to tune its parameters.**

***Index Terms*—Bayesian classifier, fuzzy classifier, fuzzy c-means (FCM), MPEG VBR video, traffic modeling, type-2 fuzzy logic systems.**

I. Introduction

MULTIMEDIA technologies will profoundly change the way we access information, conduct business, communicate, educate, learn, and entertain. Among the various kinds of multimedia services, video service is becoming an important component. Video service refers to the transmission of moving images together with sound [24]. Research on video transfers for multimedia services has been quite active in recent years, and video applications are expected to be the major source of traffic in future broad-band networks [29]. Video applications such as video on demand, automatic surveillance systems, video databases, industrial monitoring, video teleconferencing, etc., involve storage and processing of video data. Many of these applications can benefit from retrieval of the video data based on their content, but generally, any content retrieval model must have the capacity of dealing with massive amounts of data [6]. For example, how does one search for the clip "Christopher speaking at USC commencement" from a video archive consisting of an enormous number of tapes?

Digital video is often compressed by exploiting the inherent redundancies that are common in motion pictures. So, to classify the compressed (such as MPEG) video traffic directly without decompressing it will be an essential step for ensuring the effectiveness of these systems.

Most works on video sequence classification belong to the content-based approach, which uses the spatial knowledge obtained after decompressing the video sequence (e.g., [6], [31]). Research on real-time video traffic classification is scarce, because:

1) video traffic is compressed code and very little information is available for classification (the only information that can be used is temporal knowledge of the video traffic); and
2) video traffic is highly bursty and exhibits uncertain behavior.

Patel and Sethi [25] proposed a decision tree classifier for video shot detection and characterization by examining the compressed video directly. For shot detection, their method consists of comparing intensity, row, and column histograms of successive I frames of MPEG video using the χ-square test. For characterization of segmented shots, they classified shot motion into different categories using a set of features derived from motion vectors of P and B frames of MPEG video.

Relatively more research exists for video traffic characteristic modeling and predicting than for classification. Dawood and Ghanbari [4], [5] used linguistic labels to model MPEG video traffic, and classified them into nine classes based on texture and motion complexity. They used crisp values obtained from the mean values of training prototype video sequences to define low, medium, and high texture and motion. Chang and Hu [3] investigated the applications of pipelined recurrent neural networks to MPEG video traffic prediction and modeling. Intracoded (I), predicted (P), and bidirectional (B) (I/P/B) pictures were characterized by a general nonlinear ARMA process. Pancha *et al.* [26] observed that a γ distribution fits the statistical distribution of the packetized bits/frame of video traffic with low bit rates. Heyman *et al.* [10] showed that the number of bits/frame distribution of I-frames has a lognormal distribution and its autocorrelation follows a geometrical function, and they concluded that there is no specific distribution that can fit P and B frames. Krunz *et al.* [14], however, found that the lognormal distribution is the best match for all three types. All these methods belong to the statistical signal processing-based approaches, which match the mean and variance to a known statistical distribution. Recently, Krunz and Makowski [15] observed that $M/G/\infty$ input models are good candidates for modeling many types of correlated traffic (such as video traffic) in computer networks. These works are useful to us for understanding the characteristics about different video traffic so that the characteristics can be used for classification.

As noted in [22], a shortcoming to model-based statistical signal processing is "... the assumed probability model, for which model-based statistical signal processing results will be good if the data agrees with the model, but may not be so good if the data does not." In real variable bit rate (VBR) video traffic, the traffic is highly bursty, and we believe that no statistical model can really demonstrate the uncertain nature of the I/P/B

Manuscript received February 14, 2000; revised August 31, 2000. This work was supported by the Center for Research on Applied Signal Processing (CRASP) at the University of Southern California.

The authors are with the Signal and Image Processing Institute, Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089 USA (e-mail: mendel@sipi.usc.edu).

Publisher Item Identifier S 1063-6706(01)01359-5.

frames. Fuzzy logic systems (FLS) are model free. Their membership functions are not based on statistical distributions. In this paper, we, therefore, apply fuzzy techniques to MPEG VBR video traffic modeling and classification.

A survey of recent advances in fuzzy logic (FL) applied to telecommunications networks is discussed in [9]; it shows that FL is very promising for every aspect of communication networks. Recently, Tsang, Bensaou, and Lam [29] proposed a fuzzy-based real-time MPEG video rate control scheme to avoid a long delay or excessive loss at the user-network interface (UNI) in an ATM network. The success of fuzzy logic applied to communication networks motivates us to apply FL to video traffic modeling and classification. Recently, rule-based FL classifiers are reported in [16], but they are all type-1 designs. Because uncertainties are observed in video traffic modeling and classification, we shall use type-2 FL; however, we shall compare classification results obtained from classifiers based on either a type-1 or type-2 FL.

In Section II, we briefly introduce MPEG video traffic. In Section III, we introduce type-2 fuzzy sets. In Section IV, we model I/P/B frame sizes using supervised clustering and unsupervised clustering, respectively. Five fuzzy classifiers are presented in Section V. In Section VI, a Bayesian classifier is proposed. Performances of the six classifiers are evaluated in Section VII. Conclusions are presented in Section VIII.

In the sequel, we use the following notation and terminology. A is a type-1 fuzzy set, and the membership grade (a synonym for the degree of membership) of $x \in X$ in A is $\mu_{\mathrm{A}}(x)$, which is a crisp number in [0, 1]. A type-2 fuzzy set, denoted $\tilde{\mathrm{A}}$, is characterized by a type-2 membership function $\mu_{\tilde{\mathrm{A}}}(x, u)$, where $x \in X$ and $u \in J_x \subseteq [0, 1]$, i.e., $\tilde{\mathrm{A}} = \{((x, u), \mu_{\tilde{\mathrm{A}}}(x, u)) | \forall x \in X, \forall u \in J_x \subseteq J_x[0, 1]\}$, in which $0 \leq \mu_{\tilde{\mathrm{A}}}(x, u) \leq 1$. At each value of x, say $x = x'$, the 2-D plane whose axes are u and $\mu_{\tilde{\mathrm{A}}}(x'u)$ is called a vertical slice of $\mu_{\tilde{\mathrm{A}}}(x, u)$. A secondary membership function is a vertical slice of $\mu_{\tilde{\mathrm{A}}}(x, u)$. It is $\mu_{\tilde{\mathrm{A}}}(x = x', u)$ for $x' \in X$ and $\forall u \in J_{x'} \subseteq [0, 1]$, i.e., $\mu_{\tilde{\mathrm{A}}}(x = x', u) \triangleq \mu_{\tilde{\mathrm{A}}}(x') = \int_{u \in J_{x'}} f_{x'}(u)/u$, $J_{x'} \subseteq [0, 1]$. Because $\forall x' \in X$, we drop the prime notation on $\mu_{\tilde{\mathrm{A}}}(x')$, and refer to $\mu_{\tilde{\mathrm{A}}}(x)$ as a secondary membership function; it is a type-1 fuzzy set, which we also refer to as a secondary set. Based on the concept of secondary sets, we can reinterpret a type-2 fuzzy set as the union of all secondary sets, i.e., we can re-express $\tilde{\mathrm{A}}$ in a vertical-slice manner, as $\tilde{\mathrm{A}} = \{(x, \mu_{\tilde{\mathrm{A}}}(x)) | \forall x \in X\}$. The domain of a secondary membership function is called the primary membership of x; J_x is the primary membership of x, where $J_x \subseteq [0, 1]$ for $\forall x \in X$. The amplitude of a secondary membership function is called a secondary grade; $f_x(u)$ is a secondary grade. Uncertainty in the primary memberships of a type-2 fuzzy set, $\tilde{\mathrm{A}}$, consists of a bounded region called the footprint of uncertainty (FOU). It is the union of all primary memberships, i.e., $\mathrm{FOU}(\tilde{\mathrm{A}}) = \bigcup J_x$. Assume that each of the secondary membership functions of a type-2 fuzzy set has only one secondary grade that equals 1. A principal membership function is the union of all such points at which this occurs, i.e., $\mu_{\mathrm{principal}}(x) = \int_{x \in X} u/x$ where $f_x(u) = 1$, and is associated with a type-1 fuzzy set. Consider a family of type-1 membership functions $\mu_A(x|p_1, \ldots, p_v)$ where $p_1, \ldots, p_v$ are parameters, some or all of which vary over some range of values, i.e., $p_i \in P_i$ $(i = 1, \ldots, v)$. A primary membership function is any one of these membership functions, e.g., $\mu_A(x|p_1 = p_{1'}, \ldots, p_v = p_{v'})$. For short, we use $\mu_A(x)$ to denote a primary membership function. It will be subject to some restrictions on its parameters. The family of all primary membership functions creates an FOU. Meet and join are defined and explained in great detail in [12] and [13].

II. Introduction to MPEG Video Traffic

MPEG (moving picture expert group) is an ISO/IEC standard for digital video compression coding, and has been extensively used to overcome the problem of storage of prerecorded video on digital storage media, because of the high compression ratios it achieves. MPEG video traffic is composed of a group of pictures (GoP) that include encoded frames: I/P/B. I frames are coded with respect to the current frame using a two-dimensional (2-D) discrete cosine transform, and they have a relatively low compression ratio; P frames are coded with reference to previous I or P frames using interframe coding, and they can achieve a better compression ratio than I frames; and, B frames are coded with reference to the next and previous I or P frames, and B frames can achieve the highest compression ratio of the three frame types. The sequence of frames is specified by two parameters: M the distance between I and P frames; and, N the distance between I frames. The use of these three types of frames allows MPEG to be both robust (I frames permit error recovery) and efficient (B and P frames have high compression ratio). Variable bit-rate (VBR) MPEG video is used in ATM networks, and constant bit-rate (CBR) MPEG video is often used in narrowband ISDN. We focus on MPEG VBR video traffic. In Fig. 1, we plot the I/P/B frame sizes for 3000 frames of an MPEG coded video of *Star Wars*.

III. Type-2 Fuzzy Sets

The concept of type-2 fuzzy sets was introduced by Zadeh [32] as an extension of the concept of an ordinary fuzzy set, i.e., a type-1 fuzzy set [20]. Type-2 fuzzy sets have grades of membership that are themselves fuzzy [7]. A type-2 membership grade can be any subset in [0, 1]—the primary membership; and, corresponding to each primary membership, there is a secondary grade (which can also be in [0, 1]) that defines the possibilities for the primary membership. A type-1 fuzzy set is a special case of a type-2 fuzzy set; its secondary membership function is a subset with only one element, unity. Type-2 fuzzy sets allow us to handle linguistic uncertainties, as typified by the adage "words can mean different things to different people." A fuzzy relation of higher type (e.g., type-2) has been regarded as one way to increase the fuzziness of a relation, and, according to Hisdal, "increased fuzziness in a description means increased ability to handle inexact information in a logically correct manner [11]."

A general type-2 fuzzy set has lots of parameters to be determined [12], [13], but things simplify a lot when its secondary membership functions are interval sets. Interval sets are very useful when we have no other *a priori* knowledge about membership function uncertainties. In our video traffic modeling, we will focus on using interval type-2 fuzzy sets. An interval type-2

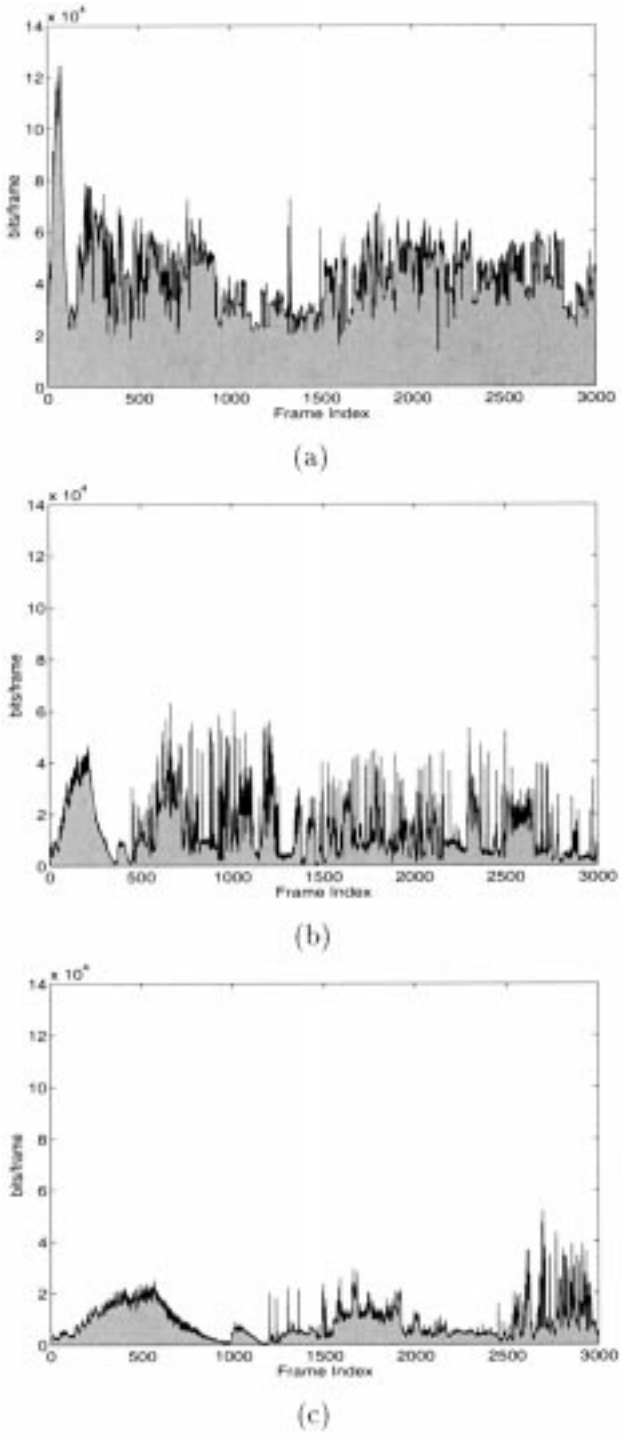

Fig. 1. Portions of I/P/B frame sizes of *Star Wars* video. (a) I frame. (b) P frame. (c) B frame.

fuzzy set can be represented by its upper and lower membership functions (MFs) [17]. An upper MF and a lower MF are two type-1 MFs, which are bounds for the FOU of an interval type-2 MF. The upper MF is associated with the upper bound of the FOU and, the lower MF is a subset which is associated with the lower bound of the FOU. Determining the footprint of uncertainty is crucial for the use of type-2 fuzzy sets, and is application dependent. We establish the FOU for MPEG coded videos in Section IV.

We use an overbar (underbar) to denote the upper (lower) MF. For example, let $\tilde{\mathrm{F}}_k^l(x_k)$ denote the type-2 MF for the kth antecedent of the lth rule, then the upper and lower MFs of $\mu_{\tilde{\mathrm{F}}_k^l}(x_k)$ are $\overline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)$ and $\underline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)$, respectively, so that

$$\mu_{\tilde{\mathrm{F}}_k^l}(x_k) = \int_{q^l \in [\underline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k),\, \overline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)]} 1/q^l \tag{1}$$

where $\int$ denotes the union of individual points of each set in the continuum.

Note that when membership function parameters are fixed, i.e., they don't vary over a range of values, then the primary membership function reduces to the type-1 membership function $\mu_A(x)$.

Example 1: Gaussian Primary MF with Uncertain Standard Deviation: Consider the case of a Gaussian primary MF having a fixed mean m_k^l and an uncertain standard deviation that takes on values in $[\sigma_{k1}^l, \sigma_{k2}^l]$, i.e.,

$$\mu_k^l(x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - m_k^l}{\sigma_k^l}\right)^2\right], \qquad \sigma_k^l \in [\sigma_{k1}^l, \sigma_{k2}^l] \tag{2}$$

where $k = 1, \ldots, p$; p is the number of antecedents; $l = 1, \ldots, M$; and, M is the number of rules. The upper MF, $\overline{\mu}_k^l(x_k)$, is (see Fig. 2)

$$\overline{\mu}_k^l(x_k) = \mathcal{N}(m_k^l, \sigma_{k2}^l; x_k) \tag{3}$$

and the lower MF, $\underline{\mu}_k^l(x_k)$, is (see Fig. 2)

$$\underline{\mu}_k^l(x_k) = \mathcal{N}(m_k^l, \sigma_{k1}^l; x_k). \tag{4}$$

□

This example illustrates how to define $\overline{\mu}$ and $\underline{\mu}$, so that it is clear how to define these membership functions for other situations (e.g., triangular, trapezoidal, bell MFs).

IV. Modeling I/P/B Frame Sizes Using Clustering

Clustering of numerical data forms the basis of many classification and modeling algorithms. The purpose of clustering is to distill natural groupings of data from a large data set, producing a concise representation of a system's behavior.

In this section, we use supervised clustering when the I/P/B frame categories are known, and unsupervised clustering when the I/P/B frame categories are unknown. Both approaches ignore the time-index for each frame, and represent the histograms of I/P/B frame sizes using three fuzzy sets, one each for I, P, and B frames. Because the I/P/B frames are mixed together in video traffic, we use clustering to group them into I, P, or B clusters. We then compute the mean and std of each cluster. These statistics are then used to determine the footprints of uncertainty for I/P/B clusters.

A. Modeling I/P/B Frame Sizes Using Supervised Clustering

To the best of our knowledge, all current approaches (e.g., [26], [10], [14]) for modeling I/P/B frame sizes belong to supervised clustering, i.e., they assume the I/P/B frame categories are known ahead of time; consequently, we shall first use a supervised clustering algorithm, assuming there are three clusters, so as to analyze the statistical nature of these clusters.

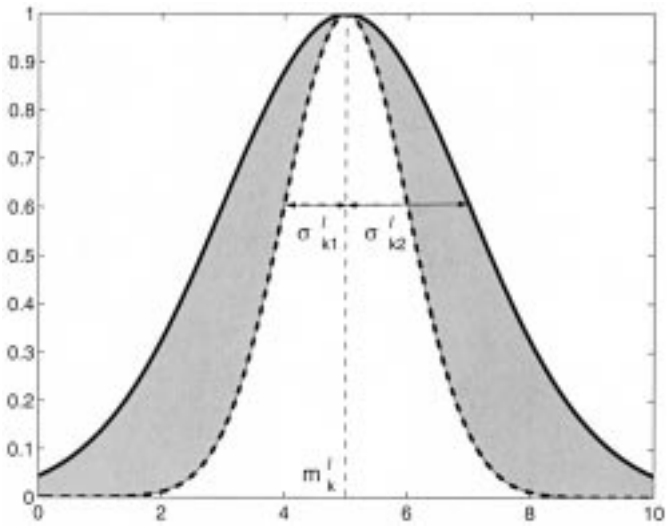

Fig. 2. The type-2 MFs for Example 1. The thick solid lines denote upper MFs, and the thick dashed lines denote lower MFs. The shaded regions are the footprints of uncertainty for interval secondaries. The center of the Gaussian MFs is 5, and the variance varies from 1.0 to 2.0.

We will show that the statistical knowledge (mean and std) about the size (bits/fame) of I, P, or B clusters is distinct for different groups of frames, even in the same video product; hence, this will motivate us to use type-2 fuzzy sets to model the number of bits/frame in I/P/B traffic, and is consistent with our belief that the frame sizes of video traffic are not really wide-sense stationary (WSS), and that their distribution varies with respect to the frame index.

Thanks to O. Rose [27] of the University of Wurzburg, who made 20 MPEG-1 video traces available on-line (FTP: //ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/), we were able to perform our experiments. Many works by others have been done based on these resources, e.g., Rose [27] analyzed their statistical properties and observed that the frame and GoP sizes can be approximated by Gamma or Lognormal distributions, Manzoni *et al*. [19] studied the workload models of VBR video traffic based on these video traces, and Adas [1] used adaptive linear prediction to forecast the VBR video for dynamic bandwidth allocation.

We used 17 of Rose's video traces, and subdivided them into four categories, movies, sports, news/talk show, and MTV, according to the subject of the video, i.e.,

- **Movies:** 1) "James Bond: Goldfinger" (*bond*), 2) "Jurassic Park" (*dino*), 3) "The Silence of the Lambs" (*lambs*), 4) "Star Wars" (*star*), 5) "Terminator II" (*term*), and 6) a 1994 movie preview (*movie*).
- **Sports:** 7) ATP tennis final (*atp*), 8) Formula 1 race: GP Hockenheim 1994 (*race*), 9) Super Bowl final 1995: San Diego—San Francisco (*sbowl*), two 1994 soccer World Cup matches [10) *soc1* and 11) *soc2*].
- **News/Talk Show:** Two German talk shows [12) *talk1* and 13) *talk2*] and two German TV news [14) *news1* and 15) *news2*].
- **MTV:** two MTV video clips [16) *mtv1* and 17) *mtv2*].

The videos were compressed by Rose using an MPEG-1 encoder using a pattern with GoP size 12, *IBBPBBPBBPBB*. Which means GoP size 12. Each MPEG video stream consisted of 40 000 video frames, which at 25 frames/s represented about 30 min of real-time full motion video. Fig. 1 shows portions of the I/P/B frame size sequences of the *star* video.

TABLE I
MEAN AND STD VALUES FOR EIGHT SEGMENTS AND THE ENTIRE *lambs* VIDEO TRAFFIC, AND THEIR NORMALIZED STD

Video Data	I Frame		P Frame		B Frame	
	mean	std	mean	std	mean	std
Segment 1	4.6478	0.1143	3.7710	0.3643	3.5080	0.2669
Segment 2	4.5563	0.1032	3.8098	0.3547	3.4643	0.3058
Segment 3	4.4990	0.0388	3.3314	0.3065	3.1011	0.2144
Segment 4	4.5087	0.0657	3.4899	0.3043	3.2489	0.2231
Segment 5	4.6538	0.1664	3.9747	0.3943	3.6660	0.3490
Segment 6	4.5407	0.1496	3.8511	0.3488	3.5359	0.3011
Segment 7	4.4739	0.1334	3.5128	0.3754	3.2645	0.3209
Segment 8	4.5907	0.1087	3.7445	0.2345	3.4798	0.1694
Entire Traffic	4.5589	0.1326	3.6857	0.3950	3.4085	0.3251
Normalized std	0.0147	0.3173	0.0590	0.1300	0.0545	0.1892

Krunz *et al.* [14] found that the lognormal distribution is the best match for all I/P/B frames, i.e., if the I, P, or B frame size at time j is s_j, then

$$\log_{10} s_j \sim \mathcal{N}(\cdot;\, m,\, \sigma^2). \tag{5}$$

We, therefore, tried to model the logarithm of the frame size, to see if a Gaussian MF can match its nature. We chose lambs, atp, talk2, and mtv2 as examples. For each video traffic, we decomposed the I/P/B frames into eight segments, and computed the mean m_i and std σ_i of the logarithm of the frame size of the ith segment, $i = 1, 2, \ldots, 8$. We also computed the mean m and std σ of the entire video traffic in a video product. To see which value—m_i or σ_i—varies more, we normalized the mean and std of each segment using m_i/m, and σ_i/σ, and we then computed the std of their normalized values σ_m and σ_{std}. As we see from the last row of Tables I–IV, $\sigma_m \ll \sigma_{std}$. We conclude, therefore, that if the I/P/B frames of each segment (short range) of the video traffic are lognormally distributed, then the logarithm of the I, P, or B frame sizes in an entire video traffic (long range) is more appropriately modeled as a Gaussian with uncertain standard deviation. This justifies the use of the Gaussian MFs, given in Example 1, to model the video traffic.

The approach in this section has been based on the assumption that the I/P/B category of a frame is known, which is a form of supervised clustering. This kind of modeling is very useful in network workload analysis (for dynamic bandwidth allocation), and network control (such as connection admission control).

In some cases, the frame category is unknown, in which case we need to use unsupervised clustering to model them.

B. Modeling I/P/B Frame Sizes Using Unsupervised Clustering

In some cases, we do not know the video traffic frame category (I, P, or B). To blindly model the statistical knowledge of I/P/B frames, we use fuzzy c-means (FCM) [2] clustering, which is an unsupervised clustering method.

TABLE II
MEAN AND STD VALUES FOR EIGHT SEGMENTS AND THE ENTIRE *atp* VIDEO TRAFFIC, AND THEIR NORMALIZED STD

Video Data	I Frame		P Frame		B Frame	
	mean	std	mean	std	mean	std
Segment 1	4.8895	0.1233	4.4542	0.2112	4.1526	0.1826
Segment 2	4.8632	0.1444	4.3765	0.2773	4.0865	0.2614
Segment 3	4.8455	0.1308	4.3185	0.2289	4.0450	0.1984
Segment 4	4.8546	0.1195	4.3619	0.1834	4.0768	0.1714
Segment 5	4.8363	0.1409	4.3169	0.2618	4.0384	0.2402
Segment 6	4.8779	0.1006	4.3647	0.1785	4.0862	0.1354
Segment 7	4.8362	0.1437	4.2942	0.2791	4.0207	0.2416
Segment 8	4.8839	0.1078	4.4349	0.2206	4.1294	0.1850
Entire Traffic	4.8610	0.1288	4.3652	0.2389	4.0795	0.2101
Normalized std	0.0044	0.1287	0.0130	0.1654	0.0111	0.2020

TABLE III
MEAN AND STD VALUES FOR EIGHT SEGMENTS AND THE ENTIRE *talk2* VIDEO TRAFFIC, AND THEIR NORMALIZED STD

Video data	I Frame		P Frame		B Frame	
	mean	std	mean	std	mean	std
Segment 1	4.8883	0.0394	4.2928	0.1114	4.0883	0.0758
Segment 2	4.8605	0.0738	4.2450	0.1133	4.0414	0.0841
Segment 3	4.8176	0.0749	4.1383	0.1901	3.9662	0.1244
Segment 4	4.8848	0.0825	4.2360	0.1880	4.0335	0.1349
Segment 5	4.8533	0.0507	4.1961	0.1375	3.9970	0.0964
Segment 6	4.8539	0.0605	4.1845	0.1435	3.9907	0.0971
Segment 7	4.8438	0.0789	4.1600	0.1425	3.9646	0.1000
Segment 8	4.8897	0.0901	4.2792	0.2452	4.0817	0.1898
Entire Traffic	4.8614	0.0745	4.2165	0.1726	4.0204	0.1263
Normalized std	0.0052	0.2307	0.0132	0.2643	0.0121	0.2904

TABLE IV
MEAN AND STD VALUES FOR EIGHT SEGMENTS AND THE ENTIRE *mtv2* VIDEO TRAFFIC, AND THEIR NORMALIZED STD

Video data	I Frame		P Frame		B Frame	
	mean	std	mean	std	mean	std
Segment 1	4.7623	0.1125	4.3718	0.2230	4.0820	0.2097
Segment 2	4.7517	0.1294	4.2415	0.3197	3.9157	0.3187
Segment 3	4.8428	0.1218	4.3807	0.2462	4.0547	0.2315
Segment 4	4.6683	0.1281	4.2511	0.2666	3.9438	0.2532
Segment 5	4.7828	0.1497	4.3012	0.3448	4.0038	0.3442
Segment 6	4.7201	0.1304	4.1269	0.3098	3.8307	0.2528
Segment 7	4.8909	0.1960	4.5381	0.3724	4.2473	0.3813
Segment 8	4.6516	0.0809	3.9668	0.2186	3.7039	0.1839
Entire Traffic	4.7590	0.1545	4.2723	0.3342	3.9727	0.3194
Normalized std	0.0171	0.2122	0.0404	0.1718	0.0416	0.2161

FCM clustering is a data clustering technique where each data point belongs to a cluster to a degree specified by a membership grade. This technique was originally introduced by Bezdek [2] as an improvement on earlier clustering methods.

The FCM method clusters the video traffic into three clusters, which we represent as three fuzzy sets by estimating the mean ($\mathbf{v}_i$) and std of the three clusters. We compute the std of the ith cluster using

$$\sigma_i^2 = \sum_{k=1}^{n} u_{ik}\, d_{ik}^2 \Big/ \sum_{k=1}^{n} u_{ik} \qquad \forall i \tag{6}$$

where

- n number of total frames;
- u_{ik} membership degree of frame k belonging to cluster i;
- d_{ik} Euclidean distance between frame k and the center of cluster i.

We assume that the clusters with the largest, next to largest and smallest mean value are I, P, and B frames, respectively. This follows from the statistical analyzes in Tables I–IV. In Fig. 3, we plot the performance of FCM clustering as compared with supervised clustering. Observe that the mean and std of the frame sizes obtained via the FCM approaches those obtained via supervised clustering; hence, when the frame categories of video traffic are unkown, FCM clustering can be used to extract the mean and std of I/P/B frames.

V. FIVE FUZZY CLASSIFIERS FOR VIDEO TRAFFIC CLASSIFICATION

In this section, we use type-1 (singleton and nonsingleton), and interval type-2 (singleton and nonsingleton) FLSs for video traffic classification. We do not attempt to develop a universal classifier that can recognize and classify any video product. Such a general-purpose (domain-independent) classifier has been shown to be too complex for present technology [6]. We assume that the domain of interest (*movies* or *sports*) is known *a priori*, and that the video classifier is confined to work only on *movie* or *sports* traffic.

We chose 10 video products from the 17 video products; five of them are *movies* (video product 2–6) and five are *sports* (video product 7–11). We wish to classify video traffic from the 10 video products as either a *movie* or *sports*, which is essentially a binary decision problem.

A. Singleton and Nonsingleton Type-1 FLSs

A FLS is described by fuzzy IF–THEN rules, which represent input–output relations of a system. For classification, in a type-1 FLS with a rule base of M rules, each having p antecedents, the lth rule R^l is expressed as

$$R^l: \quad \text{IF } x_1 \text{ is } \mathrm{F}_1^l \text{ and } x_2 \text{ is } \mathrm{F}_2^l \text{ and } \ldots \text{ and } x_p \text{ is } \mathrm{F}_p^l \quad \text{THEN } y^l = c^l$$

in which $l = 1, 2, \ldots, M$, the consequent c^l are crisp values (e.g., $+1$ or -1 in this paper), and, F_k^l $(k = 1, 2, \ldots, p)$ are type-1 fuzzy sets. For a singleton FLS, the rule firing degree is

$$f^l = \mathcal{T}_{k=1}^{p} \mu_{\mathrm{F}_k^l}(x_k) \tag{7}$$

where $\mathcal{T}$ denotes a t-norm (minimum or product); for a nonsingleton FLS [23], the rule firing degree is

$$f^l = \mathcal{T}_{k=1}^{p} \mu_{\mathrm{X}_k}(x_k) \star \mu_{\mathrm{F}_k^l}(x_k) \tag{8}$$

where x_k is fuzzified to a type-1 fuzzy set (e.g., a Gaussian centered at the measured value of x_k and with std σ).

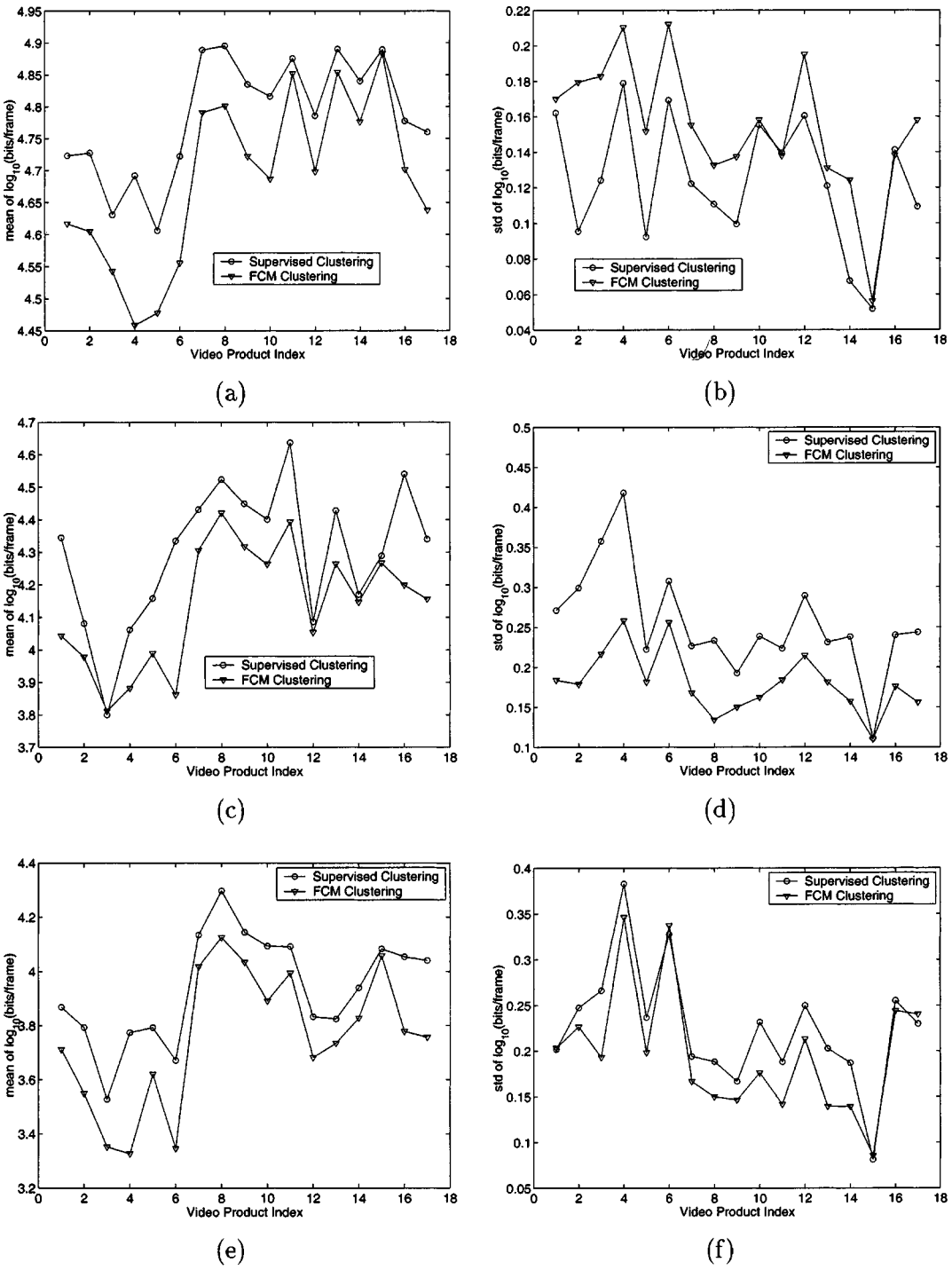

Fig. 3. Supervised and FCM clustering for the first 6000 frames of 17 video products. (a), (c), (e) are the mean of I, P, and B frame sizes ($\log_{10}$ b/frame), respectively, and (b), (d), (f) are the std of I, P, and B frame sizes ($\log_{10}$ b/frame), respectively.

In this paper, we use an unnormalized output [28] for the FLS, namely

$$y = \sum_{l=1}^{M} f^l y^l \tag{9}$$

because we make a decision based on the sign of the output, and normalization operation will not change the sign. If the domain of interest had three or more choices, we would use a normalized output for decision making.

In Section VII, we use singleton and nonsingleton type-1 FLSs for video traffic classification.

B. Singleton and Nonsingleton Interval Type-2 FLSs

In a type-2 FLS with a rule base of M rules, in which each rule has p antecedents, the lth rule R^l, is expressed as

$$R^l: \quad \text{IF } x_1 \text{ is } \tilde{\text{F}}_1^l, \text{ and } x_2 \text{ is } \tilde{\text{F}}_2^l, \text{ and }, \ldots, \text{ and } x_p \text{ is } \tilde{\text{F}}_p^l \quad \text{THEN } y^l = c^l.$$

The firing degree F^l of rule R^l is

$$F^l = \sqcup_{\mathbf{x}\in\mathbf{X}}\Big\{\Big[\mu_{\tilde{\mathrm{X}}_1}(x_1)\sqcap\mu_{\tilde{\mathrm{F}}_1^l}(x_1)\Big] \sqcap\cdots\sqcap\Big[\mu_{\tilde{\mathrm{X}}_p}(x_p)\sqcap\mu_{\tilde{\mathrm{F}}_p^l}(x_p)\Big]\Big\} \tag{10}$$

where $\mathbf{X} = X_1 \times X_2 \times \cdots \times X_p$ is the measurement domain of $\mathbf{x} = [x_1, x_2, \ldots, x_p]$. General type-2 FLSs are computationally intensive because it is not easy to evaluate F^l in (10). Things simplify a lot when secondary membership functions are interval sets, in which case secondary memberships are either 0 or 1, and, as we demonstrate later, such simplifications make the use of a type-2 FLS practical.

1) Operations Between Input and Antecedent: We can have three type-2 FLSs, each based on a different fuzzification of the input, i.e., input fuzzified to a type-2 set, input fuzzified to a type-1 set, and input fuzzified to a type-0 set (i.e., singleton fuzzification). Results that are needed about these three kinds of type-2 FLSs are summarized next, because in Section VII we use them all for video traffic classification.

Theorem 1 [17]: In an interval type-2 nonsingleton FLS with type-2 fuzzification and meet under minimum or product t-norm, the result of the input and antecedent operations, $F^l(\mathbf{x}')$ in (10), is an interval type-1 set, i.e., $F^l(\mathbf{x}') = [\underline{f}^l(\mathbf{x}'), \overline{f}^l(\mathbf{x}')]$, where

$$\underline{f}^l(\mathbf{x}') = \sup_{\mathbf{x}} \int_{x_1\in X_1}\cdots\int_{x_p\in X_p}\Big[\underline{\mu}_{\tilde{\mathrm{X}}_1}(x_1)\star\underline{\mu}_{\tilde{F}_1^l}(x_1)\Big] \star\cdots\star\Big[\underline{\mu}_{\tilde{\mathrm{X}}_p}(x_p)\star\underline{\mu}_{\tilde{F}_p^l}(x_p)\Big]\Big/\mathbf{x} \tag{11}$$

and

$$\overline{f}^l(\mathbf{x}') = \sup_{\mathbf{x}} \int_{x_1\in X_1}\cdots\int_{x_p\in X_p}\Big[\overline{\mu}_{\tilde{\mathrm{X}}_1}(x_1)\star\overline{\mu}_{\tilde{F}_1^l}(x_1)\Big] \star\cdots\star\Big[\overline{\mu}_{\tilde{\mathrm{X}}_p}(x_p)\star\overline{\mu}_{\tilde{F}_p^l}(x_p)\Big]\Big/\mathbf{x}; \tag{12}$$

the supremum is attained when each term in brackets attains its supremum.

The proof of this Theorem is given in [17].

Because $\underline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)\star\underline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)$ and $\overline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)\star\overline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)$ only depend on x_k, the supremum in (11) and (12) is attained when each term in brackets attains its supremum; hence, in the inference of a type-2 FLS, we will examine

$$\overline{f}_k^l(x_k') \triangleq \sup_{x_k}\int_{x_k\in X_k}\Big[\overline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)\star\overline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)\Big]\Big/x_k \tag{13}$$

$$\underline{f}_k^l(x_k') \triangleq \sup_{x_k}\int_{x_k\in X_k}\Big[\underline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)\star\underline{\mu}_{\tilde{\mathrm{F}}_k^l}(x_k)\Big]\Big/x_k \tag{14}$$

where $k = 1, \ldots, p$, and $\star$ is a t-norm; then, $\overline{f}^l$ and $\underline{f}^l$ can be reexpressed, as

$$\overline{f}^l = \mathcal{T}_{k=1}^p\overline{f}_k^l \tag{15}$$

$$\underline{f}^l = \mathcal{T}_{k=1}^p\underline{f}_k^l \tag{16}$$

where $\mathcal{T}$ denotes t-norm.

When the input is fuzzified to a type-1 fuzzy set, so that $\mu_{\tilde{\mathrm{X}}_k}\to\mu_{\mathrm{X}_k}$ $(k = 1, \ldots, p)$, the upper and lower MFs of $\mu_{\tilde{\mathrm{X}}_k}$ merge into one MF, $\mu_{\mathrm{X}_k}(x_k)$, in which case Theorem 1 simplifies to:

Corollary 1: In an interval type-2 FLS with *nonsingleton type-1 fuzzification* and meet under minimum or product t-norm, $\underline{f}^l$ and $\overline{f}^l$, in (11) and (12), simplify to

$$\underline{f}^l(\mathbf{x}') = \sup_{\mathbf{x}} \int_{x_1\in X_1}\cdots\int_{x_p\in X_p}\Big[\mu_{X_1}(x_1)\star\underline{\mu}_{\tilde{F}_1^l}(x_1)\Big] \star\cdots\star\Big[\mu_{X_p}(x_p)\star\underline{\mu}_{\tilde{F}_p^l}(x_p)\Big]\Big/\mathbf{x} \tag{17}$$

and

$$\overline{f}^l(\mathbf{x}') = \sup_{\mathbf{x}} \int_{x_1\in X_1}\cdots\int_{x_p\in X_p}\Big[\mu_{X_1}(x_1)\star\overline{\mu}_{\tilde{F}_1^l}(x_1)\Big] \star\cdots\star\Big[\mu_{X_p}(x_p)\star\overline{\mu}_{\tilde{F}_p^l}(x_p)\Big]\Big/\mathbf{x} \tag{18}$$

where μ_{X_k} $(k = 1, \ldots, p)$ is the type-1 fuzzified input.

When a singleton fuzzifier is used, the upper and lower MFs of $\mu_{\tilde{\mathrm{X}}_k}(x_k)$ merge into one crisp value, namely 1, in which case Theorem 1 simplifies further to:

Corollary 2: In an interval type-2 FLS with *singleton fuzzification* and meet under minimum or product t-norm, $\underline{f}^l$ and $\overline{f}^l$, in (11) and (12), simplify to

$$\underline{f}^l = \underline{\mu}_{\tilde{F}_1^l}(x_1)\star\cdots\star\underline{\mu}_{\tilde{F}_p^l}(x_p) \tag{19}$$

and

$$\overline{f}^l = \overline{\mu}_{\tilde{F}_1^l}(x_1)\star\cdots\star\overline{\mu}_{\tilde{F}_p^l}(x_p) \tag{20}$$

where x_i $(i = 1, \ldots, p)$ denotes the location of the singleton.

Example 1: Input and Antecedents are Gaussian Primary MFs with Uncertain Standard Deviations: In this example, we compute $\overline{f}_k^l$ and $\underline{f}_k^l$ when both the input fuzzy sets and antecedent MFs are Gaussian primary MFs with uncertain standard deviations. In this case

$$\mu_{\tilde{\mathrm{X}}_k}(x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - m_{\tilde{\mathrm{X}}_k}}{\sigma_{\tilde{\mathrm{X}}_k}}\right)^2\right], \quad \sigma_{\tilde{\mathrm{X}}_k}\in\left[\sigma_{\tilde{\mathrm{X}}_{k1}}, \sigma_{\tilde{\mathrm{X}}_{k2}}\right] \tag{21}$$

and, its upper and lower MFs are $\overline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)$ and $\underline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)$, respectively. The kth antecedent MF has the following form:

$$\mu_{\tilde{\mathrm{F}}_k^l}(x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - m_{\tilde{\mathrm{F}}_k^i}}{\sigma_{\tilde{\mathrm{F}}_k^i}}\right)^2\right], \quad \sigma_{\tilde{\mathrm{F}}_k^i}\in\left[\sigma_{\tilde{\mathrm{F}}_{k1}^i}, \sigma_{\tilde{\mathrm{F}}_{k2}^i}\right] \tag{22}$$

and, its upper and lower MFs are $\overline{\mu}_{\tilde{\mathrm{F}}_l^k}(x_k)$ and $\underline{\mu}_{\tilde{\mathrm{F}}_l^k}(x_k)$, respectively.

Our type-2 input and antecedent membership functions are depicted in Fig. 4. Observe, from (13), that $\overline{f}_k^l$ requires the calculation of the supremum of a product or minimum of the two Gaussians $\overline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)$ and $\overline{\mu}_{\tilde{\mathrm{F}}_l^k}(x_k)$. Such a calculation has been carried out by Mouzouris and Mendel in [23] for the type-1 nonsingleton case; they derive the value of x_k at which $\sup_{x_k\in X_k}\int_{X_k}[\mu_{\mathrm{X}_k}(x_k)\star\mu_{\mathrm{F}_k^l}(x_k)]/x_k$ is achieved. Denoting

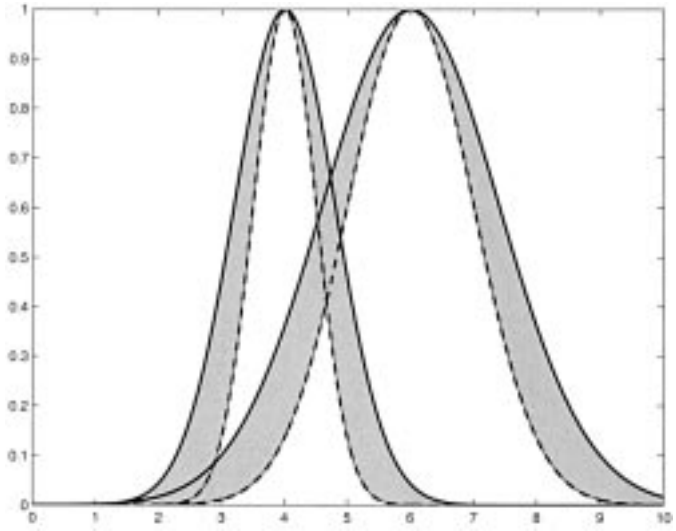

Fig. 4. The type-2 MFs for Example 2. The MF with the larger variance is the antecedent MF, and the other is the input MF. The thick solid lines denote upper MFs, and the thick dashed lines denote lower MFs. The shaded regions are the footprints of uncertainty.

this value of x_k as $x_{k,\max}^l$, they have shown that, for product t-norm

$$x_{k,\max}^l = \frac{\sigma_{\tilde{\mathrm{X}}_k}^2 m_{\tilde{\mathrm{F}}_k^l} + \sigma_{\tilde{\mathrm{F}}_k^l}^2 x_k'}{\sigma_{\tilde{\mathrm{X}}_k}^2 + \sigma_{\tilde{\mathrm{F}}_k^l}^2}. \tag{23}$$

Denoting the value of x_k at which the supremum of (13) occurs as $\overline{x}_{k,\max}^l$, it follows from (23) and Fig. 4, that for product t-norm

$$\overline{x}_{k,\max}^l = \frac{\sigma_{\tilde{\mathrm{X}}_{k2}}^2 m_{\tilde{\mathrm{F}}_k^i} + \sigma_{\tilde{\mathrm{F}}_{k2}^i}^2 x_k'}{\sigma_{\tilde{\mathrm{X}}_{k2}}^2 + \sigma_{\tilde{\mathrm{F}}_{k2}^i{}^2}^2}. \tag{24}$$

Similarly, $\underline{f}_k^l$ requires the calculation of the supremum of a product or minimum of the two Gaussians $\underline{\mu}_{\tilde{\mathrm{X}}_k}(x_k)$ and $\underline{\mu}_{\tilde{\mathrm{F}}_l^k}(x_k)$. Denoting the value of x_k at which this supremum occurs as $\underline{x}_{k,\max}^l$, it again follows from (23) and Fig. 4, that for product t-norm

$$\underline{x}_{k,\max}^l = \frac{\sigma_{\tilde{\mathrm{X}}_{k1}}^2 m_{\tilde{\mathrm{F}}_k^i} + \sigma_{\tilde{\mathrm{F}}_{k1}^i}^2 x_k'}{\sigma_{\tilde{\mathrm{X}}_{k1}}^2 + \sigma_{\tilde{\mathrm{F}}_{k1}^i}^2}. \tag{25}$$

From these results, it is straightforward to compute $\overline{f}_k^l$ and $\underline{f}_k^l$ using (13) and (14), i.e.,

$$\begin{aligned}\overline{f}_k^l(x_k') &= \overline{\mu}_{\tilde{\mathrm{X}}_k}\left(\overline{x}_{k,\max}^l\right)\overline{\mu}_{\tilde{\mathrm{F}}_k^l}\left(\overline{x}_{k,\max}^l\right)\\ &= \exp\left[-\frac{1}{2}\left(\frac{\overline{x}_{k,\max}^l - x_k'}{\sigma_{\tilde{\mathrm{X}}_{k2}}}\right)^2\right]\\ &\quad\cdot\exp\left[-\frac{1}{2}\left(\frac{\overline{x}_{k,\max}^l - m_{\tilde{\mathrm{F}}_k^i}}{\sigma_{\tilde{\mathrm{F}}_{k2}^i}}\right)^2\right] \qquad (26)\\ &= \exp\left[-\frac{1}{2}\frac{\left(m_{\tilde{\mathrm{F}}_k^i} - x_k'\right)^2}{\sigma_{\tilde{\mathrm{X}}_{k2}}^2 + \sigma_{\tilde{\mathrm{F}}_k^i}^2}\right] \qquad (27)\end{aligned}$$

$$\begin{aligned}\underline{f}_k^l(x_k') &= \underline{\mu}_{\tilde{\mathrm{X}}_k}\left(\underline{x}_{k,\max}^l\right)\underline{\mu}_{\tilde{F}_k^l}\left(\underline{x}_{k,\max}^l\right)\\ &= \exp\left[-\frac{1}{2}\left(\frac{\underline{x}_{k,\max}^l - x_k'}{\sigma_{\tilde{\mathrm{X}}_{k1}}}\right)^2\right]\\ &\quad\cdot\exp\left[-\frac{1}{2}\left(\frac{\underline{x}_{k,\max}^l - m_{\tilde{\mathrm{F}}_k^i}}{\sigma_{\tilde{\mathrm{F}}_k^i}}\right)^2\right] \qquad (28)\\ &= \exp\left[-\frac{1}{2}\frac{\left(m_{\tilde{\mathrm{F}}_k^i} - x_k'\right)^2}{\sigma_{\tilde{\mathrm{X}}_{k1}}^2 + \sigma_{\tilde{\mathrm{F}}_{k1}^i}^2}\right]. \qquad (29)\end{aligned}$$

The results in this example are needed later in our type-2 FCs.

2) Output Processing: The final output of the type-2 FLS is obtained by applying the Extension Principle [32] to (9), i.e.,

$$\begin{aligned}&Y(\mathbf{x}')\\ &= \int_{f^1\in[\underline{f}^1,\overline{f}^1]}\cdots\int_{f^M\in[\underline{f}^M,\overline{f}^M]} \mathcal{T}_{i=1}^M \mu_{F^i}(f^i) \Big/ \sum_{i=1}^M f^i y^i.\end{aligned} \tag{30}$$

Because F^i ($i = 1, 2, \ldots, M$) are interval type-1 sets, i.e., $\mu_{\mathrm{F}^i}(f^i) = 1$, (30) simplifies to

$$\begin{aligned}Y(\mathbf{x}') &= \int_{f^1\in[\underline{f}^1,\overline{f}^1]}\cdots\int_{f^M\in[\underline{f}^M,\overline{f}^M]} 1 \Big/ \sum_{i=1}^M f^i y^i\\ &= [y_l, y_r].\end{aligned} \tag{31}$$

Because $f^i \in F^i = [\underline{f}^i, \overline{f}^i]$ and y^i is a crisp value, we obtain [18]

$$y_r = \sum_{l=1}^M \overline{f}^l y^l \tag{32}$$

$$y_l = \sum_{l=1}^M \underline{f}^l y^l. \tag{33}$$

The defuzzified output of the type-2 FLS is $y = (y_l + y_r)/2$, i.e.,

$$y = \sum_{l=1}^M y^l \left(\underline{f}^l + \overline{f}^l\right)\Big/ 2. \tag{34}$$

In order to compute y in (34), we just need to compute $\underline{f}^l$ and $\overline{f}^l$ based on Theorem 1, Corollary 1 or Corollary 2. □

We have presented five FLSs, two are type-1 and three are type-2. In Section VII, we use these five FLSs as fuzzy classifiers for video traffic classification. Our fuzzy classifiers are model-free. To compare them against a model-based classifier, we develop a Bayesian classifier in Section VI.

VI. Bayesian Classifier for Video Traffic Classification

Bayesian decision theory [8] provides the optimal solution to the general decision-making problem. We assume that each video product v_i is equiprobable, i.e., $p(v_i) = 1/10$, where $i \in \{2, \ldots, 11\}$ (e.g., $i = 2$ corresponds to the *movie* Jurassic

Park). Let H_1: *movie*, and H_2: *sports*, so $p(H_1) = p(H_2) = 0.5$. If each component of the frame size, $\mathbf{s}_i \triangleq [s_i^I, s_i^P, s_i^B]^T$ is a lognormal function [14] of the I, P, and B frames of the ith video product, $i = 2, \ldots, 11$, and $\mathbf{x}_i \triangleq \log \mathbf{s}_i$, then

$$p(\mathbf{x}_i|v_i) = \frac{1}{(2\pi)^{3/2}|\Sigma_\mathbf{i}|^{1/2}} \cdot \exp\left[-\frac{1}{2}(\mathbf{x}_i - \mathbf{m}_i)^T \Sigma_\mathbf{i}^{-1}(\mathbf{x}_i - \mathbf{m}_i)\right] \quad (35)$$

where $\mathbf{m}_i \triangleq [m_i^I, m_i^P, m_i^B]^T$ and $\Sigma_\mathbf{i} = \text{diag}\{\sigma_i^{I^2}, \sigma_i^{P^2}, \sigma_i^{B^2}\}$ are the mean vector (3×1) and covariance matrix (3×3) of $\mathbf{x}_i$. In this case

$$p(\mathbf{x}|H_1) = \sum_{i=2}^{6} p(\mathbf{x}|v_i)p(v_i) \quad (36)$$

$$p(\mathbf{x}|H_2) = \sum_{i=7}^{11} p(\mathbf{x}|v_i)p(v_i). \quad (37)$$

Based on Bayes decision theory, since $p(H_1) = p(H_2) = 0.5$, we obtain the decision rule:

The video traffic is $movie$ if

$$p(\mathbf{x}|H_1) > p(\mathbf{x}|H_2) \quad (38)$$

The video traffic is $sports$ if

$$p(\mathbf{x}|H_1) < p(\mathbf{x}|H_2) \quad (39)$$

The video traffic is $movie$ or $sports$ if

$$p(\mathbf{x}|H_1) = p(\mathbf{x}|H_2). \quad (40)$$

This Bayesian classifier will be used in Section VII.

VII. Simulations

We extract the general features and behavior of MPEG video traffic for different video products, and determine one discriminant rule for each kind of traffic in the domain of interest. In choosing the antecedents of the fuzzy classifier, we make full use of the statistical knowledge (mean and std) obtained from the video traffic.

For the ten video products we chose, we used their first 24 000 frames for supervised clustering to establish a discriminant rule for that video product. All in all, we obtained 10 rules, one per product.

To evaluate the performance of the five FC, we used the next 10 000 (24 001–34 000 frames, 500×20) frames for (in-product) testing, i.e., for classifying a video as a *movie* or *sport*. We tested every 500 frames, with 20 independent evaluations for each video product. During testing, we used FCM clustering to obtain the mean $\mathbf{m^t} = [m_I^t, m_P^t, m_B^t]$ and std $\sigma^\mathbf{t} = [\sigma_I^t, \sigma_P^t, \sigma_B^t]$ of I/P/B frames for each 500 frames.

We also describe two other experiments performed to test the robustness of the classifiers when the testing video product is *not* included in the training video products (i.e., out-of-product testing).

A. Design of Six Video Classifiers

1) Design of Type-1 Fuzzy Classifiers: For a type-1 fuzzy classifier, the lth rule, R^l, is ($l = 1, \ldots, 10$):

R^l: IF I frame is F_1^l and P frame is F_2^l and B frame is F_3^l THEN this product is movie $(+1)$ [or sports(-1)].

The antecedents F_k^l ($k = 1, 2, 3$) are described by a type-1 Gaussian MF whose mean, m_p^l, and std, σ_p^l, are determined by supervised clustering. More specifically, m_1^l and σ_1^l are the mean and std of all I frames in the first 24 000 frames of video product l; m_2^l and σ_2^l are the mean and std of all P frames in the first 24 000 frames of video product l; and, m_3^l and σ_3^l are the mean and std of all B frames in the first 24 000 frames of video product l. The consequent $y^l = c^l$ in Section V-A corresponds to $y^l = 1$ or $y^l = -1$ in the fuzzy classifier.

For a type-1 singleton FC (type-1 SFC), its input $\mathbf{m^t}$ is obtained from FCM clustering. For a type-1 nonsingleton FC (type-1 NSFC), its three inputs are each type-1 Gaussians with mean $[m_I^t, m_P^t, m_B^t]$ and std $[\sigma_I^t, \sigma_P^t, \sigma_B^t]$, which are obtained from FCM clustering (6).

2) Design of Type-2 Fuzzy Classifiers: For type-2 fuzzy classifiers, the lth rule, R^l, is ($l = 1, \ldots, 10$):

R^l: IF I frame is $\tilde{\text{F}}_1^l$ and P frame is $\tilde{\text{F}}_2^l$ and B frame is $\tilde{\text{F}}_3^l$ THEN this product is movie $(+1)$ [or sports (-1)].

The antecedents $\tilde{\text{F}}_k^l$ ($k = 1, 2, 3$) are described by a type-2 MF, i.e., a Gaussian MF with uncertain std, whose mean m_k^l and std $\sigma_k^l \in [\sigma_{k1}^l, \sigma_{k2}^l]$ are determined by supervised clustering. More specifically, m_k^l ($k = 1, 2, 3$) are determined using the same method as described in Section VII-A.1, and σ_{k1}^l and σ_{k2}^l are determined as follows. We divided the 24 000 frames of the lth product into five equal-length (4800 frames) segments, and computed the std σ_1^{lj} of all I frames in the jth segment ($j = 1, \ldots, 5$). Let

$$\sigma_{11}^l = \min_{j=1,\ldots,5} \sigma_1^{lj} \quad (41)$$

$$\sigma_{12}^l = \max_{j=1,\ldots,5} \sigma_1^{lj} \quad (42)$$

so $[\sigma_{11}^l, \sigma_{12}^l]$ is the range of uncertain std of I frames of the lth video product. We obtained the ranges of uncertain standard deviations of P frames ($[\sigma_{21}^l, \sigma_{22}^l]$) and B frames ($[\sigma_{31}^l, \sigma_{32}^l]$) in a similar manner.

For a type-2 singleton FC (type-2 SFC), its input, $\mathbf{m^t} = [m_I^t, m_P^t, m_B^t]$, is also obtained from FCM. For a type-2 nonsingleton FC with type-1 input (type-2 NSFC-1), its input is three type-1 Gaussians with mean $[m_I^t, m_P^t, m_B^t]$ and std $[\sigma_I^t, \sigma_P^t, \sigma_B^t]$, which are obtained from FCM. For the type-2 nonsingleton FC with type-2 input (type-2 NSFC-2), its inputs are three type-2 MFs (Gaussian MFs with mean $[m_I^t, m_P^t, m_B^t]$ and uncertain std $[\sigma_I^t, \sigma_P^t, \sigma_B^t]$). To set the uncertain std $\sigma^\mathbf{t}$ we divided each testing segment (500 frames) into four subsegments (125 frames), and used FCM clustering for each subsegment; then, the minimum and maximum values of the four standard deviations were used to set the range of uncertain $[\sigma_I^t, \sigma_P^t, \sigma_B^t]$.

TABLE V
FALSE ALARM RATE (FAR) OF SIX CLASSIFIERS

Classifiers	FAR in Experiment I	FAR in Experiment II	FAR in Experiment III
Bayesian Classifier	11.5%	14.29%	14.29%
type-1 SFC	10.5%	15.07%	9.41%
type-1 NSFC	8.5%	14.35%	9.17%
type-2 SFC	11.5%	14.24%	13.65%
type-2 NSFC-1	7%	14.11%	8.43%
type-2 NSFC-2	8.5%	14.35%	8.03%

TABLE VI
PARAMETERS OF THE FIVE FCS THAT WERE TUNED AND THE STEP SIZE USED IN THEIR STEEPEST DESCENT TUNING ALGORITHM. THE CONSEQUENT c^l IS INITIALLY SET $+1$ FOR MOVIE AND -1 FOR SPORTS; $k = 1, 2, 3$ AND $l = 1, 2, \ldots, 8$

Classifiers	Antecedents	Consequents	Step Size α	Number of parameters tuned
type-1 SFC	m_k^l, σ_k^l	c^l	10^{-4}	56
type-1 NSFC	m_k^l, σ_k^l	c^l	10^{-4}	56
type-2 SFC	m_k^l, $[\sigma_{k1}^l, \sigma_{k2}^l]$	c^l	5×10^{-7}	80
type-2 NSFC-1	m_k^l, $[\sigma_{k1}^l, \sigma_{k2}^l]$	c^l	2×10^{-3}	80
type-2 NSFC-2	m_k^l, $[\sigma_{k1}^l, \sigma_{k2}^l]$	c^l	10^{-3}	80

3) Design of Bayesian Classifier: Observe from (35), that the Bayesian classifier needs $\mathbf{m}_i = [m_i^I, m_i^P, m_i^B]^T$ and $\Sigma_\mathbf{i} = \text{diag}\{\sigma_i^{I^2}, \sigma_i^{P^2}, \sigma_i^{B^2}\}$. In our design, m_i^I and σ_i^I are the mean and std of all I frames in the first 24 000 frames of video product i; m_i^P and σ_i^P are the mean and std of all P frames in the first 24 000 frames of video product i; and, m_i^B and σ_i^B are the mean and std of all B frames in the first 24 000 frames of video product i; and, its input $\mathbf{x} \triangleq \mathbf{m^t}$, where $\mathbf{m^t}$ is obtained from FCM.

B. Performance

We computed the average (in-product) false-alarm rate (FAR) for each fuzzy classifier, as well as for the Bayesian classifier in $20 \times 10 = 200$ independent classifications (10 video traffic products each with 20 500-frame segments).

The average FARs of these six classifiers in this experiment (Experiment I are summarized in Table V. From this table, we observe that a type-2 NSFC-1 performs the best. The reason why a type-2 NSFC-2 did not perform the best is because, for a short range of frames (500 frames), a type-1 input Gaussian MF is enough to capture its uncertainty. The type-2 NSFC-1 outperformed the Bayesian classifier by $(11.5 - 7)/(11.5) = 39.13\%$, because apparently a stationary lognormal distribution is not so accurate for modeling the frame sizes of MPEG video traffic.

C. Robustness

To examine the robustness of the six classifiers, we designed them using eight video traffic, four from video products 2–6 (*movies*), and four from video products 7–11 (*sports*), and tested their performance using the two unused video products, one from video products 2–6, and one from video products 7–11. So we totally have 25 independent combinations—eight video products for training plus two video products for (out-of-product) testing.

1) Robustness Testing without Parameter Adjustments: We used the first 24 000 frames of each of the eight training video products to establish a discriminant rule for that video product using the methods described in Sections VII-A.1 and 2. All in all, we obtained eight rules, one per product. We evaluated the performance of the six classifiers using the first 37 500 (500×75) frames of the two (out-of-product) testing videos, i.e., for classifying a video as a *movie* or *sport*. We tested every 500 frames, with 75 independent evaluations for each video product, so we totally evaluated the six classifiers $25 \times 2 \times 75 = 3750$ times. The average FARs of these 6 classifiers for 3750 decisions (Experiment II) are summarized in Table V. From Table V, we see that the performance of all classifiers are about the same, which calls into question the use of a FC over the Bayesian classifier. So far, though, we have not optimized the parameters of the FCs. We do this next, and then reexamine robustness.

2) Robustness Testing with Parameter Adjustments: The FCs have many parameters that must be specified, as summarized in Table VI. Here we tune the parameters using the training data and a steepest descent algorithm. Parameter adjustments using steepest-descent algorithm for type-1 FLSs have been extensively used (e.g., [30]). A steepest-descent algorithm for tuning a type-2 FLS was proposed in [17].

Initially, we used the first 24 000 frames of each of the eight training video products to establish a discriminant rule for that video product using the methods described in Sections VII-A.1 and 2. All in all, we obtained eight rules, one per product. We divided the 24 000 frames of each video product into 40 groups, 600 frames per group. We extracted the mean $\mathbf{m}$ and std σ for the I/P/B frames in each group using supervised clustering. For all SFCs, we used $\mathbf{m}$ as the input, and the video product category (*movie*, $+1$; and *sports*, -1) as the desired output. For type-1 NSFC and type-2 NSFC-1, we used Gaussian MFs with mean $\mathbf{m}$ and std σ as the input, and the video product category as the desired output. For type-2 NSFC, we used Gaussian MFs with mean $\mathbf{m}$ and uncertain std (obtained by computing σ for

each sub-group) as the input, and the video product category as the desired output. So for each FC, we obtained $40 \times 8 = 320$ training prototypes. We used a steepest descent algorithm to tune the five fuzzy classifiers. The parameters that were tuned and the step size α of the 5 FCs are summarized in Table VI. We tried our best to choose the step size so that every FC achieved its best performance.

After tuning, we fixed the parameters in each fuzzy classifier, and performed the same out-of-product testing as described in Section VII-C-2. The average FAR of these five fuzzy classifiers for 3750 decisions (Experiment III) are summarized in Table V. From Table V, we see that the *type-2 NSFC-2* performs the best. It outperforms the Bayesian classifier (in Experiment II) by $(14.29 - 8.03)/(14.29) = 43.81\%$, which shows that our tuned FCs are robust.

VIII. Conclusion and Future Work

We have proposed two ways for modeling I/P/B frame sizes of video traffic, using either type-1 or type-2 fuzzy sets. We observed that for long-range video traffic, a Gaussian MF with uncertain std is appropriate for modeling the frame sizes. We also used FCM to cluster and model the frame sizes when the frame category is unknown.

We have classified video traffic using compressed data and have proposed type-2 FLCs to do this. Five fuzzy classifiers have been used for video traffic classification, and have shown that:

1) When the testing video product is included in the training products (but the testing frames are not included in the training frames), and design of the classifiers is based only on the statistical knowledge of training video traffic (i.e., none of the classifier's parameters are tuned), the type-2 NSFC-1 performs the best.
2) When the testing video product is not included in the training products, and the classifier's parameters are tuned using steepest descent algorithm, the type-2 NSFC-2 performs the best.

We have classified a video product as a *movie* or *sport*, which is essentially a binary detection problem. How to classify a video product in a larger domain (such as four possible choices) and keep a low FAR is our future research.

References

[1] A. M. Adas, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model," *IEEE Trans. Networking*, vol. 6, pp. 635–644, Oct. 1998.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[3] P.-R. Chang and J.-T. Hu, "Optimal nonlinear adaptive prediction and modeling of MPEG video in ATM networks using pipelined recurrent neural networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 1087–1100, Aug. 1997.

[4] A. M. Dawood and M. Ghanbari, "MPEG video modeling based on scene description," in *IEEE Int. Conf. Image Processing*, vol. 2, Chicago, IL, Oct. 1998, pp. 351–355.

[5] ——, "Content-based MPEG video traffic modeling," *IEEE Trans. Multimedia*, vol. 1, pp. 77–87, Mar. 1999.

[6] N. Dimitrova and F. Golshani, "Motion recovery for video content classification," *ACM Trans. Inform. Syst.*, vol. 13, no. 4, pp. 408–439, Oct. 1995.

[7] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. New York: Academic, 1980.

[8] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[9] S. Ghosh, Q. Razouqi, H. J. Schumacher, and A. Celmins, "A survey of recent advances in fuzzy logic in telecommunications networks and new challenges," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 443–447, Aug. 1998.

[10] D. P. Heyman, A. Tabatabi, and T. V. Lakshman, "Statistical analysis of MPEG-2 coded VBR video traffic," in *6th Int. Workshop on Packet Video*, Portland, OR, Sept. 1994.

[11] E. Hisdal, "The IF THEN ELSE statement and interval-valued fuzzy sets of higher type," *Int. J. Man-Machine Studies*, vol. 15, pp. 385–455, 1981.

[12] N. N. Karnik and J. M. Mendel. (1998, Oct.) An Introduction to Type-2 Fuzzy Logic Systems. USC Rep. [Online]. Available: http://sipi.usc.edu/~mendel/report

[13] N. N. Karnik, J. M. Mendel, and Q. Liang, "Type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 643–658, Dec. 1999.

[14] M. Krunz, R. Sass, and H. Hughes, "Statistical characteristics and multiplexing of MPEG streams," in *Proc. IEEE Int. Conf. Comput. Commun., INFOCOM'95*, vol. 2, Boston, MA, Apr. 1995, pp. 445–462.

[15] M. Krunz and A. M. Makowski, "Modeling video traffic using $M/G/\infty$ input processes: A compromise between Markovian and LRD models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 733–748, June 1998.

[16] K. I. Kuncheva, *Fuzzy Classifier Design*. Heidelberg: Physica-Verlag, 2000.

[17] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 535–550, Oct. 2000.

[18] ——, "Interval type-2 TSK fuzzy logic systems, with application to MPEG VBR video traffic forecasting," IEEE Trans. Syst., Man, Cybern., to be published.

[19] P. Manzoni, P. Cremonesi, and G. Serazzi, "Workload models of VBR video traffic and their use in resource allocation policies," *IEEE Trans. Networking*, vol. 7, pp. 387–397, June 1999.

[20] J. M. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, pp. 345–377, Mar. 1995.

[21] ——, "Computing with words when words can mean different things to different people," in *Int. ICSC Congress on Computational Intelligence: Methods Applicat., Third Ann. Symp. Fuzzy Logic and Applicat.*, Rochester, NY, June 22–25, 1999.

[22] ——, "Uncertainty, fuzzy logic, and signal processing," *Signal Processing*, vol. 80, no. 6, pp. 913–933, June 2000.

[23] G. C. Mouzouris and J. M. Mendel, "Nonsingleton fuzzy logic systems: Theory and application," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 56–71, Feb. 1997.

[24] G. Pacifici, G. Karlsson, M. Garrett, and N. Ohta, "Guest editorial real-time video services in multimedia networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 961–964, Aug. 1997.

[25] N. Patel and I. K. Sethi, "Video shot detection and characterization for video databases," *Pattern Recog.*, vol. 30, no. 4, pp. 583–592, 1997.

[26] P. Pancha and M. El-Zarki, "A look at the MPEG video coding standard for variable bit rate video transmission," in *Proc.IEEE INFOCOM'92*, Florence, Italy, 1992.

[27] O. Rose, "Satistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," Univ. Wurzburg, Inst. Comput. Sci., Rep. 101, Feb. 1995.

[28] K. Tanaka, M. Sano, and H. Watanabe, "Modeling and control of carbon monoxide concentration using a neuro-fuzzy technique," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 271–279, Aug. 1995.

[29] D. Tsang, B. Bensaou, and S. Lam, "Fuzzy-based rate control for real-time MPEG video," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 504–516, Nov. 1998.

[30] L.-X. Wang, *Adaptive Fuzzy Systems and Control*. Englewood Cliffs, NJ: PTR Prentice-Hall, 1994.

[31] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying production effects," *Multimedia Systems*, vol. 7, pp. 119–128, 1999.

[32] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inform. Sci.*, vol. 8, pp. 199–249, 1975.