Multimedia Systems 7: 199–213 (1999)

Multimedia Systems
© Springer-Verlag 1999

# A survey of statistical source models for variable-bit-rate compressed video

**Michael R. Izquierdo**[1], **Douglas S. Reeves**[2]

[1] Multimedia Access Corp., Osprey Technologies, Cary, NC 27511, USA
[2] North Carolina State University, Raleigh, NC 27695, USA

**Abstract.** It is predicted that, in the near future, the transport of compressed video will pervade computer networks. Variable-bit-rate (VBR) encoded video is expected to become a significant source of network traffic, due to its advantages in statistical multiplexing gain and consistent video quality. Both systems analysts and developers need to assess and study the impact these sources will have on their networks and networking products. To this end, suitable statistical source models are required to analyze performance metrics such as packet loss, delay and jitter. This paper provides a survey of VBR source models which can be used to drive network simulations. The models are categorized into four groups: Markov chain/linear regression, TES, self-similar and i.i.d/analytical. We present models which have been used for VBR sources containing moderate-to-significant scene changes and moderate-to-full motion. A description of each model is given along with corresponding advantages and shortcomings. Comparisons are made based on the complexity of each model.

**Key words:** Video modeling – VBR – Variable bit rate – MPEG – H.261 – TES – Self-similar

## 1 Introduction

The recent development of standards for digital video compression, such as H.261 [1], H.263, [2] MPEG-1 [3] and MPEG-2 [4], has made it feasible to transport video data over computer communications networks. It is predicted that in the near future, transporting video over computer networks will become commonplace. Classical models based on a renewal procees (e.g., Poisson process), traditionally used in the analysis of telephony networks, are not adequate to model video traffic. This is due to the fact that the Poisson process assumes that arrivals are independent, whereas, for compressed video, they are not. In consequence, new models are needed to describe compressed video sources, and derive attendant performance measures.

Compressed video is a traffic source which can have high peak-to-mean ratios and significantly high autocorrelations. This type of source can be deleterious to networks, since it can cause severe data loss if network resources are not properly allocated. One way to ameliorate this difficulty is to control the output bit rate of the encoder. This is referred to as constant-bit-rate (CBR) encoding. A CBR encoder's output bit rate is nearly constant, making it possible to transport its output using a fixed-rate channel. This makes bandwidth allocation simpler and also renders the video source more amenable to traffic policing. However, CBR encoding has the drawback that video quality (distortion) varies significantly in order to maintain a constant bit rate. Also, the desired bit rate needs to be determined up front, which for some applications, such as the real-time encoding of a live sporting event, might not be optimal since a worst case bit rate might be chosen.

An alternative to CBR encoding is variable-bit-rate (VBR) encoding. VBR does not attempt to control the output bit rate of the encoder, so distortion[1] does not vary significantly. One way to accomplish this is to keep the quantizer step size fixed. However, this makes the output bit rate of VBR encoders vary considerably, making bandwidth allocation difficult. On the other hand, this variability increases the opportunity for improvements in statistical multiplexing gain (SMG[2]). Figure 1 depicts the basic trade-off between distortion versus bit rate for compressed video. We observe that distortion varies as an encoder controls the bit rate (CBR), but if the encoder controls distortion (VBR), then bit rate varies. This behavior highlights the basic differences between CBR and VBR encoding.

Increases in SMG occurs due to the fact that the likelihood of multiple bursty sources simultaneously transmitting at their peak bit rates is small. One of the first investigations regarding the SMG of encoded video was done by Haskell, who found that a 2:1 gain was achievable when multiplex-

*Correspondence to*: M.R. Izquierdo

[1] Distortion is a coarse measure of video quality and is measured in terms of signal-to-noise ratio (SNR).

[2] In the literature, two definitions of SMG have arisen. One is the ratio fo the multiplexer output link utilization for $N$ VBR sources compared to only one source, given the same packet (cell) loss probability. The second defines SMG as the ratio of the number of VBR sources to the number of CBR sources (preferably encoded using the same source material; however, this is often not the case and is usualy estimated) which a multiplexer can accommodate given the same link capacity

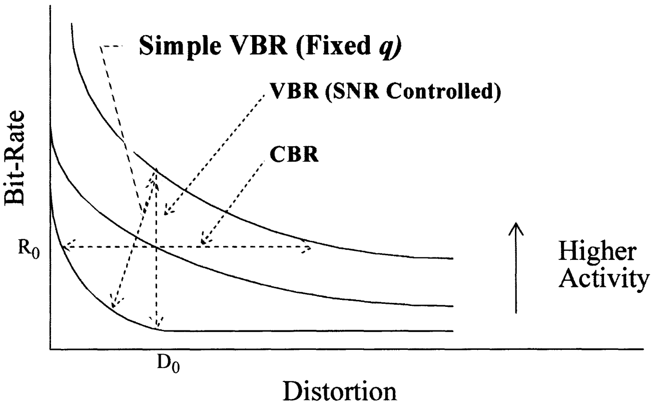


**Fig. 1.** Distortion versus bit-rate curves for compressed video

ing the outputs of several AT&T *Picturephone*® encoders [5]. The SMG of VBR sources has been further quantified in early works by Kishino et al. [6], Morrison [7] and Verbiest et al. [8], where gains of up to 4:1 were obtained for simple sequences.

One drawback to multiplexing VBR sources is the possibility of packet loss. Such losses result when multiple independent VBR sources cause the multiplexing buffer to overflow. Analyzing the multiplexer buffer occupancy behavior is a major concern for researchers. Simulation studies are often used in order to quantify the amount of loss, since analytic methods are often intractable. In order to run these simulations, source models are used to provide input stimulus to the system under study. While actual video traces may be used in place of a source model, this limits the input to a finite realization of the underlying stochastic process, which reduces the generality of the simulation results. In addition, it is often difficult to acquire video traces for long video sequences. For example, trace files for long MPEG-2 sequences are not readily available.

In this paper, we survey statistical source models for VBR video which have been proposed for both video conferencing and movie sequences. We define video conference models as those being encoded using either H.261 or MPEG without $B$ frames. A movie sequence is one which is encoded using MPEG with $I$, $B$ and $P$ frames. Some examples of models presented for movie sequences are: *Star Wars,*[3] *Last Action Hero,*[4] and *The Wizard of Oz.*[5] Only MPEG-1 models were covered in this survey since a viable MPEG-2 model had not been published by the time this survey was concluded. For introductory material on the encoding standards H.261 and MPEG-1, refer to [9] and [10], respectively.

The motivation underlying the choice of these particular models is to present a representative sampling of current VBR source models. The models are grouped into four categories: AR/Markov, TES, self-similar, analytical/i.i.d. This categorization was motivated by the dominate stochastic process used in the model. The AR/Markov models are considered a classical approach to modeling, while TES and self-similar are considered to be more novel. We provide detailed descriptions of each model, where necessary, and discuss the motivations and merits of each model. We also describe how each model was validated and the number of parameters used by the model.

The paper is organized as follows. Section 2 discusses statistical modeling and model validation. Section 3 covers models based on Markov chains and linear regressive processes. Section 4 covers models based on the TES process. Section 5 describes models based on self-similar processes. Section 6 presents analytical and non-Markovian models which are based on i.i.d. processes. Section 7 summarizes the paper, including a comparison of the number of parameters required by each model in this survey. Recommendations and issues relating to model selection are also discussed, as well as suggestions for further research in VBR modeling.

## 2 Statistical modeling and model validation

A good model is one which can accurately predict performance measures (statistics) of a stochastic system. For example, if the researcher is interested in the cell loss probability for an ATM buffer with VBR sources, then a good source model is one which produces a sample path which can accurately predict this performance measure, when the system is simulated. In most cases the validity or "goodness" of a model is determined by comparing model predictions (e.g., simulation statistics using the empirical data as the traffic source) and the corresponding statistics using the model as the traffic source. It is possible for a model to predict one metric accurately and another inaccurately. For example, a model may provide accurate predictions for cell loss probability, and be inaccurate predicting mean cell delay. The researcher must decide beforehand what the desired system metrics should be, and select a model which can accurately predict these metrics.

There are reasons to argue that the validation of a source model requires that its distribution and autocorrelation function match well their empirical counterparts, while using as few parameters as possible[6]. Typically, one tries to fit the empirical data with a classical distribution such as lognormal or Gamma, but whether or not a good fit is found, the empirical distribution (say, histogram) is a good fit by default. A common method used to match distributions is the QQ plot. Matching the autocorrelation function is a more difficult task. The autocorrelation function is a proxy for the temporal (linear) dependence within a stochastic process. Generally, stochastic processes may be classified into three types: independent, short-range dependent (SRD) and long-range dependent (LRD). An independent source is always uncorrelated, i.e., is identically zero for positive lags; the converse is false, namely, lack of correlation does not imply independence. If the autocorrelation function is summable (e.g., when it decays exponentially fast), then it is referred to as an SRD process, but if it is not summable (e.g., when it decays hyperbolically), then the source is referred to as an LRD process. The requirement to use few parameters is motivated by the fact that they must be estimated from the empirical data. Each estimate incurs a certain amount of error which tends to reduce the accuracy of the model as the

[3] Lucasfilm Ltd./20th Century Fox, 1977.

[4] Columbia Pictures, 1993

[5] MGM, 1939.

[6] Exact matching (at all lags) of the autocorrelation function is a subject of ongoing debate (see Heyman and Lakshman [11])

number of parameters increases (although those errors may occasionally cancel out). In this survey, we classify the models into two types: hierarchical and non-hierarchical. Models which capture scene changes explicitly are referred to as hierarchical models. A scene change process aims to model the relative frequency of individual scene types over longer time scale (minutes or hours) than the bit-rate process (milliseconds). Scene changes occur when the mean of the bit-rate process changes significantly as a result of a considerable change in picture content (camera cuts).

The parsimony of a model is determined by the number of parameters it requires and its complexity by the amount of computer time and memory required to generate a sample. Models which require many parameters generally require many calculations in order to generate a sample, but there are exceptions (e.g., TES). On the other hand, some models require few parameters, but take a long time to generate each sample, since each sample is calculated from all previous samples (e.g., LRD and self-similar models). It is desirable to develop a model of minimal complexity which provides sufficiently accurate predictions of the metrics of interest.

## 3 Models based on Markov chains/linear regressive processes

We provide a brief review of Markov chains and linear regressive processes, such as the autoregressive (AR) process, since many of the VBR source models are based on them. Both processes incorporate temporal dependence. AR processes, in most case, use Gaussian random variables, producing sequences which are normally distributed. Markov chains, those achieving steady state, can produce a wide variety of distributions.

### *3.1 Review of Markov process*

Models based on a Markov process use states to represent bit-rate regimes (roughly a range of bit rates of a video sequence). A stochastic process $\{Xk\}$, $k = 1, 2, \ldots$ with state space $S = \{1, 2, 3, \ldots\}$ is Markovian if for every $n$ and all states $i_1, i_2, \ldots$, where $i_n \in S$ it satisfies the Markov property,

$$\begin{aligned} P[X_n|X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, \ldots, X_1 = i_2] \\ = P[X_n = i_n|X_{n-1} = i_{n-1}]. \end{aligned} \tag{1}$$

Simply put, the current state of a Markov process depends only on its previous state, and not on any additional previous states. A stochastic process is called a Markov chain if the state space is countably infinite or finite.

Markov chains are often used to modulate other processes such as Bernoulli, Poisson or AR. The state of the Markov chain represents a different set of parameters for the particular process. While in a particular state, the model generates samples according to the particular process (Bernoulli, etc.), at the specific parameter settings. This is done for a period of time until the process switches to a different state, generating samples using a different set of parameters. Models of this type are referred to as *Markov-modulated* or *Markov-modified* models. Some examples of such models are the Markov-modulated Bernoulli process (MMBP) and the Markov-modulated Poisson process (MMPP). We will show many examples of video models which use Markov modulation.

### *3.2 Review of linear regressive process*

In an AR process, the current value is a function of a weighted linear combination of past values. Formally, it is expressed as

$$x(n) = a_0 + \sum_{i=1}^{p} a_i(x)(n - i) + e(n), \tag{2}$$

where $a_0$ is called the intercept and $\{a_1, a_2, \ldots, a_p\}$ are AR coefficients, $p$ is the order of the AR process and $\{e(n)\}$ are the residuals, commonly assumed uncorrelated and normally distributed. The AR coefficients can be determined using the recursive Levinson-Durbin algorithm [12, Appendix 2A]. AR processes of order $p$ are denoted by $AR(p)$. A special case of Eq. 2 is the AR(1) process

$$x(n) = a_0 + a_1 x(n - 1) + e(n), \tag{3}$$

where $a_1$ is the autocorrelation coefficient at lag-1 when the sequence is stationary. Note that Eq. 3 can be seen as a continuous-state, discrete-time Markov process. A model of this form was presented in [13].

The AR process is a special case of the autoregressive moving-average process (ARMA) which adds a moving-average process (MA), giving

$$x(n) = a_0 + \sum_{i=1}^{p} a_i x(n - 1) + \sum_{j=0}^{p} b_j e(n). \tag{4}$$

The ARMA process is typically stated as ARMA(p,q), where $p$ is the order of the AR part and $q$ is the order of the MA part. Determining the coefficients, $b_j$, is a bit more involved than an AR process and usually requires some form of spectral analysis [14].

### *3.3 ATM cell-level model using ARMA*

Grunenfelder et al. [14] developed a model, from a 4-s video conference sequence, for the ATM cell interarrival process from a video encoder using conditional replenishment[7]. The model defined a fixed time interval of 64 slots, where 1 slot equaled the time to transmit a 36-byte cell. The random process, $\{X_i\}$, defined the number of cells generated by the encoder within this interval, where

$$\begin{aligned} X_i &= g(\alpha Z_{i-m} + Y_i + \nu_i), \quad |\alpha < 1 \\ Y_i &= \sum_{k=-m/2}^{m/2} h_k \varepsilon_{i-k}, \end{aligned} \tag{5}$$

where $\nu_i$ is white noise. The parameters for the model were estimated from the long-term mean, variance and autocovariance of the empirical sequence. The coefficients, $h_k$, of

[7] Encoders using conditional replenishment only transmit the difference in the pixel areas between a reference and current frame. This is done using differential pulse code modulation (DPCM). Areas which do not change are run-length coded (RLC). When this difference becomes excessive, a new reference frame is generated and transmitted

the MA part were determined using Fourier analysis. The ARMA process was referred to as a *colored-Gaussian* process with zero-mean, unit variance which implies that the autocorrelation function is not from a pure Gaussian process.

In digital signal processing terms, the ARMA sequence is generated using the ARMA filter with white noise as input. Since the original sequence is not a zero-mean process, the output of the filter, $\{V_i\}$, is transformed using a zero-mean non-linearity (ZMNL) function, g(.), of the form $aV_i + b$.

This model requires 10,003 parameters, $(\alpha, a, b, h_1, h_2, \ldots h_{10,000})$, where the MA coefficients cover approximately seven frames. Model parameters were estimated from four seconds of video. This model can be viewed as modeling the video sequence at the sub-frame layer (slice/group of blocks) and it matches the pseudo-periodic autocorrelation function, typical of these sequences, quite well.

### 3.4 Video conference model using Markov chain

Heyman et al. [15] developed a frame-level[8] ATM model of a 30-min video conference sequence with no scene changes and moderate motion. The model defines the number of ATM cells per frame, $X_n$, and the state of the Markov chain, $Y_n$, where $Y_n = \lfloor X_n/10 \rfloor$[9]. The transition probability matrix, $P = [p_{ij}]$, was estimated using

$$p_{ij} = \frac{\text{number of transitions from state } i \text{ to state } j}{\text{number of transitions out of state } i}. \tag{6}$$

This model requires many parameters due to the transition probability matrix. In order to reduce the number of parameters, the authors used the DAR(1) process, which estimates the transition probabilities by using the empirical marginal distribution and autocorrelation coefficient. The transition matrix is given by

$$P = \rho I + (1 - \rho)Q, \tag{7}$$

where $\rho$ is the autocorrelation at lag 1, $I$ is the identity matrix, and each row of $Q$ consists of the marginal probability distribution function (pdf) of the empirical data. Since the empirical data was found to fit a negative-binomial distribution, each row in $Q$ contained the probabilities $(f_0, f_1, \ldots, f_K, f_K^c)$ defined by

$$\begin{aligned} f_k &= \binom{k+r+1}{k} p^r (1-p)^k, \\ f_K^c &= \sum_{k>K} f_k, \end{aligned} \tag{8}$$

where the parameters $r$ and $p$ are estimated from the empirical data, and $K$ is the maximum number of cells in a frame. By using DAR, the number of parameters was reduced to only four: peak, mean, variance, and autocorrelation at lag 1.

Interestingly, this model gave rise to better bit-rate predictions than a second-order AR process, which was also proposed at the time. This model is suitable for video conferences with no significant scene changes, since it does not model the scene changes explicitly. The model does rely on the use of a classical distribution (negative-binomial) in DAR; however, the empirical distribution could be used.

Lucantoni et al. [16] proposed a model using a discrete-state, continuous-time Markov renewal process (MRP), which they compared to the DAR model. This model is of the same vein as MMPP, but instead the bit-rate is fixed, not probabilistic. They divided the range of possible rates into 40 equidistant levels and assigned a state in the Markov chain to each level. One, and sometimes two, geometric distribution were fitted to sojourn time at each level. Sample paths generated by the model are strikingly similar in appearance to the empirical trace data. They compared the leaky bucket contour curves, generated using both models, and found that MRP was better than DAR in approximating the results produced when using the empirical sequence.

### 3.5 Hierarchical model using composite AR processes and Markov chain

Ramamurthy and Sengupta [17] proposed a hierarchical model which uses a Markov chain to capture the effects of a scene change. The model consists of two AR processes, where the first attempts to match the autocorrelation function at short lags and the second, the autocorrelation at long lags. The third process is a Markov process which is used for scene changes. Combining the three processes gives the final model

$$T_i = X_i + Y_i + Z_i, \tag{9}$$

where

$$X_i = a_1 X_{i-1} + A_i, \tag{10}$$

$$Y_i = a_2 Y_{i-1} + B_i \quad \text{and} \tag{11}$$

$$Z_i = K_i C_i. \tag{12}$$

Equation 10 is used to generate a sequence whose autocorrelation function matches that of the emprical sequence at short lags, while Eq. 11 matches it at long lags. Both equations are AR(1) processes, where $A_i$ and $B_i$ are normally distributed with means: $\mu_1$, $\mu_2$ and standard deviations: $\sigma_1$, $\sigma_2$. Equation 12 is used to generate the extra bits needed when a scene change occurs, where $K_i$ represents the state of the Markov chain and $C_i$ is a normally distributed random variable whose mean and variance depends on $K_i$. The number of parameters required for $C_i$ is reduced from four to two by making the mean and variance a function of $\alpha$ and $\beta$, where $\mu = \alpha$, $\sigma = \beta$ when $K_i = 2$ and $\mu = \alpha/2$, $\sigma = \beta/2$ when $K_i = 1$. The basic premise behind the use of the Markov chain, shown in Fig. 2, is to generate the extra bits needed in the two frames following a scene change. We will see later a model proposed by Heyman and Lakshman [21] which also takes into consideration the first two frames after a scene change. This model requires a total of eight parameters $(\alpha_1, \mu_1, \sigma_1, \alpha_2, \mu_2, \sigma_2, \alpha, \beta)$.

### 3.6 MPEG frame and slice layer models using Markov chains

Pancha and El Zarki [18] proposed an MPEG-frame and slice-layer Markov chain model for a 3-min 40-s sequence

[8] The frame level corresponds to MPEG or H.261 pictures.

[9] $\lfloor x \rfloor$ is the floor function, where $x$ is rounded towards $-\infty$

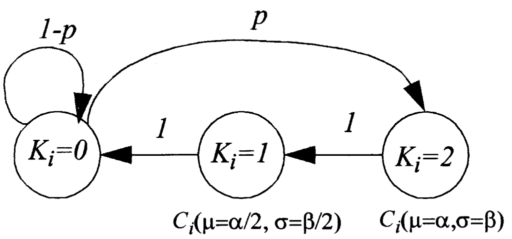


**Fig. 2.** Markov chain of scene change process

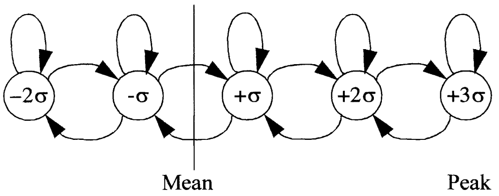


**Fig. 3.** Markov chain model for video sources

of the movie *Star Wars*. This model differs from Heyman and Lakshman's [21] model in that, rather than each state representing the number of cells in a frame, each state represents a bit-rate change of one standard deviation. This is illustrated in Fig. 3. Transition matrices were given for different group of pictures (GOP) sizes and, in general, the larger the GOP size, the more states required in the Markov chain. Also, the slice layer required more states than the frame layer. The number of parameters required by the models ranged from 51 for the frame layer to 102 for the slice layer, where the mean and standard deviation was estimated from the empirical trace data and added to the number of transition probabilities which we summarize in Table 1.

### *3.7 Video conference model using composite AR processes*

Yegenoglu et al. [19] analyzed a full-motion color video sequence of 500 frames encoded by discrete cosine transform (DCT), differential pulse code modulation (DPCM) and motion compensation. The picture resolution was 720 × 480 pixels with 16 bits per pixel and the rate was 30 frames per second. The primary motivation of this model was to produce a multimodal probability density function (in this case, Gaussian). Previous work indicated that the probability density functions of VBR video conference streams appeared to consist of a combination of probability distribution functions; indeed, the model did produce a multimodal probability density function.

The model is based on an AR process whose parameters are modulated by a Markov chain. The model quantizes bit rates into $N$ levels, where a quantization level loosely corresponds to a scene class. The Markov chain defines the transition process between quantization levels, where a single distinct AR(1) process is defined for each state of the Markov chain. When the bit rate crosses a quantization level, the first frame of the new quantization level is sampled from an *i.i.d.* Gaussian random variable.

**Table 1.** Transition probability count for frame and slice layer models

| | | Transition probabilities | |
|---|---|---|---|
| GOP Size ($N$) | Frame type | Frame | Slice |
| 1 | I | 49 | 64 |
| 16 | I,P | 64 | 81 |
| 2688 | I | 64 | 100 |

The state of the Markov chain at a particular time instant, $t$, is defined as $x_t$, where $x_t \neq x_{t-1}$ signifies a state change has occurred. For each state, $i$, there are a unique set of coefficients which is used to determine the number of bits per frame, $y_t$, given by the regressive relation

$$y_t = \begin{cases} a(i)y_{t-q} + G(\mu(i), \sigma(i)^2), & \text{if } x_t = x_{t-1} = i \\ G(\eta(i), \nu(i)), & \text{if } x_t \neq x_{t-1};\ x_t = i \end{cases} \tag{13}$$

where $G(.)$ is a Gaussian random variable and $a(i)$ is the autocorrelation coefficient at lag 1 at state $i$. When a state transition occurs, a sample is drawn from $G(\eta(i), \nu(i))$, where the mean and variance of the bit-rate process is conditioned on the state of the Markov chain, $i$. The parameters for Eq. 13 are calculated using the following constraints

$$a(i) = 1 - \frac{D^2(i)}{2\nu(i)}, \tag{14}$$

$$\mu(i) = \frac{\eta(i)(D^2(i))}{2\nu(i)}, \tag{15}$$

$$\sigma(i)^2 = D^2(i)\left(1 - \frac{D^2(i)}{4\nu(i)}\right), \tag{16}$$

where

$$D^2(i) = E[(y_t - y_{t-1})^2 | x_t = x_{t-1} = i] \tag{17}$$

is the conditioned expected square difference of the bit rates in adjacent frames. This parameter allows for the empirical data to be characterized in one pass, rather than the two passes it would have otherwise required. The values $\eta(i)$, $\nu(i)$ and $D^2(i)$ are estimated from the empirical data.

The probability density function, $f_Y(y)$, of the number of bits per frame, $y_t$, is approximated by a combination of $N$ Gaussian densities,

$$f_Y(y) = \sum_{i=1}^{N} p_i f_{G(\eta(i),\nu(i))}(y), \tag{18}$$

where $p_i$ is the steady-state probability of state $i$ in the underlying Markov chain.

Three states were used to represent the quantizations (0,44), (44,55) and (55,∞) kbits/s, resulting in good agreement between the distributions of the model and empirical data using the Kolmogorov-Smirnov test. The model captured the first, second and fourth moments of the actual data; however, the third moment in one particular state differed significantly. This was due to the fact that the actual data was not symmetrical about the mean beyond 55 kbits/s.

This model requires a total of nine parameters, three for each quantization level $\{D^2(i), \eta(i), \nu(i)\}$, where $i = \{1, 2, 3\}$. It is useful for video conference sequences with small-to-moderate motion and scene changes. The tricky part is to determine the appropriate quantization levels, a task that requires visual inspection of the empirical bit-rate distribution.

### 3.8 Block-based video conference model using AR processes

Jabbari et al. [20] developed a block-based bit-rate model of a video encoder which adhered to the general MPEG syntax. The encoder differs from the recommended MPEG implementation in that the resolution is CCIR (720 × 480 pixels/frame) instead of SIF (360 × 240 pixels/field). Interlaced scanning is used, producing two fields for every frame. In most aspects, the encoder is more similar to MPEG-2 than to MPEG-1. The statistics of encoded blocks contained within $I$ and $P$ frames were estimated for a sequence of 350 fields (175 frames); the model does not account for B frames. The model divides each field into three block types: 0, 1 and 2, where a block is defined as an 8 × 8 matrix of pixels. Type-0 blocks are encoded using DPCM and are used for sequences with very little motion. Type-1 blocks use motion compensation on sequences containing moderate motion. Type-2 blocks use DCT on sequences with high motion. $I$ frames contain only type-2 blocks, while $P$ frames contain all three types.

The model consists of a vector-valued sequence

$$\begin{aligned} x_0(n) &= a_0 x_0(n-1) + e_0(n), \\ x_1(n) &= a_1 x_1(n-1) + b_0 x_0(n) + b_2 x_2(n) + e_1(n), \qquad (19) \\ x_2(n) &= a_2 x_2(n-1) + e_2(n), \end{aligned}$$

whose components represent the number of type-0, -1 and -2 blocks in each frame, $n$ is the frame index and $e_i(n)$ is a Gaussian random variable with mean $\mu_i$ and variance $\sigma_i^2$ for $i = 0, 1$ and 2. The determination of the parameters for Eq. 19 is involved and is not included here; but details can be found in [20].

The average number of bits for block type $i$, within frame $n$, is given by

$$r_i(n) = c_i r_i(n-1) + g_i(n), \qquad (20)$$

where $c_i$ is the first-order AR coefficient and $g_i$ is a Gaussian random variable. This is used to find the total bits per block type $i$ within frame $n$ given as

$$u_i(n) = r_i(n) x_i(n). \quad i = 0, 1 \text{ and } 2. \qquad (21)$$

The final model for the total number of bits per frame, $u_T(n)$, is therefore,

$$u_T(n) = u_0(n) + u_i(n) + u_2(n). \qquad (22)$$

The model produced sample data which was shown to match the distribution of the actual data well via a QQ plot. However, the model was not used in the simulation of a multiplexer and the autocorrelation function of $u_T$ was not given. A total of 20 parameters are required for this model, of which

$$\{a_0, \mu_0, \sigma_0, a_1, b_0, b_2, \mu_1, \sigma_1, a_2, \mu_2, \sigma_2\}$$

are used for the number blocks per frame, and $c_i, \mu_i, \sigma_i$), $i = \{0, 1, 2\}$, are used for the number of bits per block.

**Table 2.** Process summary of hierarchical model

| Process Name | Process Type | Parameters |
|---|---|---|
| Scene Length | i.i.d | $\alpha, \lambda$ |
| Number of cells in scene change frame | i.i.d | $\alpha, \lambda$ |
| Number of cells in next frame<br>after scene change frame | deterministic | $\alpha, \beta, \sigma^2$ |
| Number of cells in frame within a scene | correlated | $\rho, r, p$ |

### 3.9 Hierarchical model using the DAR process

Heyman and Lakshman [21] proposed an ATM frame-layer model for sequences generated using a DPCM codec (motion compensation was not used) consisting of three different stochastic processes: (1) scene length, (2) size of the first frame after a scene change, and (3) size of frames within a scene. Scene change boundaries were determined using the second difference

$$\Delta_i = \frac{(X_{i+1} - X_i) - (X_i - X_{i-1})}{\frac{1}{m}\sum_{k=1}^{m} X_{i-k}}, \qquad (23)$$

where $X_i$ is the number of bits in frame $i$. A scene change is defined to occur when $\Delta_i$ is large and negative ($m$ was heuristically determined). The scene change process was determined to be uncorrelated; consequently, matching the distribution was sufficient. The first frame after a scene change was found to follow a different stochastic process since the scene content completely changes causing the frame to be regenerated. It was found that scene length distributions fit Weibull, Gamma and Pareto distributions well, while the number of cells within a scene change frame fit Weibull, Gamma, and in one case normal distributions (some sequences, however, did not appear to fit any distribution). The frame following a scene change frame was generated using linear prediction of the form

$$Y_i = a + bX_i + \varepsilon_i, \qquad (24)$$

where $a$ and $b$ are fixed coefficients and $\varepsilon_\iota$ is white noise.

Since the frame size within a scene is correlated, a Markov chain was used where each state represented the integer part of $X_i/50$ and the transition probability matrix was parsimoniously determined using DAR. Each row of the matrix Q consisted of the negative-binomial probabilities of the process $\{X_i/50\}$ (*refer to Sect. 3.4*). The final model requires a total of ten parameters, which we summarize in Table 2.

Frater et al. [22] proposed a similar model; however, the scene change boundaries were determined in a slightly different way and processes for the first two frames after a scene change were not considered. Scene change boundaries were determined by detecting a large difference in the output of combination median, averaging filters. The scene length pdf was found to be of the form

$$p(x) = \frac{a}{x^n + b^2}, \qquad (25)$$

where $a$ and $b$ are constants, and $n$ is estimated from the empirical sequence.

This model also used the DAR process and the negative-binomial distribution for the frame size process; thereby, requiring a total of five parameters ($\rho, r, p, b, n$). Even though the scene change frame process was not modeled, cell loss simulation results did appear to match well the results produced when using empirical trace data.

## 4 Models based on the TES process

TES processes are designed to fit simultaneously both the marginal distribution and the autocorrelation function of the empirical data. It is a general method in that it can match any marginal distribution exactly and simultaneously approximate a wide variety of autocorrelation functions. A software package called *TEStool* supports $TES$ modeling via a graphical user interface, which facilitates both algorithms and interactive searches for TES models [23]. Recently, Jelenkovic and Melamed [24] have developed an algorithm which largely automates the search process and invariably leads to a more accurate model than those obtained via human interaction.

TES's ability to match marginal distributions exactly and approximate many autocorrelation functions makes it an excellent choice for constructing a source model. We give a brief overview of TES processes; however, for more in-depth information, the reader is referred to [25].

### *4.1 Overview of TES process*

$TES$ defines a method for generating an auxiliary *background process*, $\{U_n\}$, which allows one to vary the nature of dependence among the target random variables $\{X_n\}$. The process $\{X_n\}$ is referred to as the foreground process and is generated from $\{U_n\}$ by using a suitable transformation. The process $\{U_n\}$ defines a random walk on the unit circle (circumference 1) based on the modulo-1 (fractional part) operator, defined as $\langle x \rangle = x - \lfloor x \rfloor$.

Specifically, $TES$ background processes come in two flavors, $\{U^+\}$ and $\{U_n^-\}$, defined as,

$$U_n^+ = \begin{cases} U_0, & n = 0 \\ \langle U_{n-1}^+ + V_n \rangle, & n > 0 \end{cases} \quad U_n^- = \begin{cases} U_n^+, & n\ even \\ 1 - U_n^+, & n\ odd \end{cases} \tag{26}$$

where the initial value $U_0$ is uniformly distributed on the interval $[0, 1)$ and $\{V_n\}$, called the *innovation sequence*, consists of a sequence of i.i.d. random variables which are independent of $U_0$. The background process $\{U^+\}$ can generate both negative decaying or oscillatory autocorrelation functions, while $\{U^-\}$ generates autocorrelations which alternate in polarity between odd and even lags.

In general, $\{Vn\}$ is obtained from the innovation density $f_V$, which is typically restricted to the class of step-function innovation densities over the interval $[-0.5, 0.5)$ in order to simplify the parameter search. In the simplest non-trivial case of uniform innovations, $\{V_n\}$ is determined by two parameters $L$ and $R$, where

$$V_n = L + (R - L)Z_n \tag{27}$$

is generally referred to as the *single-step innovation function*, $Z_n$ are i.i.d. and uniformly distributed on the interval $[0, 1)$ and $-0.5 \le L < R < 0.5$. The parametrization $(L, R)$ is equivalent to the parametrization $(\alpha, \phi)$, where

$$\begin{aligned} \alpha &= R - L, \\ \phi &= \frac{R + L}{R - L}. \end{aligned} \tag{28}$$

The $(\alpha, \phi)$ parametrization is convenient for calculating the autocorrelation function. The parameter a controls the magnitude of the autocorrelation function and $\phi$ controls its oscillations.

Once the background process is determined, the next step is to define the desired *foreground process* $\{X_n\}$. This is done by applying a transformation called the *distortion function*, to $\{U_n^+\}$ or $\{U_n^-\}$. A common distortion is of the form $D = H_Y^{-1} \circ S_\xi$, where $H_Y^{-1}$ is the inverse of the cumulative histogram of the empirical data and $S_\xi$ is a stitching transform. A *stitching transformation* is used, parametrized by a *stitching parameter* $0 \le \xi \le 1$, where

$$S_\xi(y) = \begin{cases} \dfrac{y}{\xi}, & 0 \le y \le \xi \\ \dfrac{1 - y}{1 - \xi}, & \xi \le y \le 1 \end{cases} \tag{29}$$

$S_\xi$ is an intermediate step designed to "smooth" the sample paths of $\{U_n^+\}$ and $\{U_n^-\}$ when crossing the origin. In most cases, a value at or near 0.5 is selected for $\xi$. Finally, the requisite foreground TES sequence is defined by,

$$X_n = H_Y^{-1}(S_\xi(U_n)). \tag{30}$$

### *4.2 H.261 GOB video conference model using a TES process*

Melamed et al. [26] developed a model for the number of bits per GOB for H.261-encoded video. In H.261, each frame (352 × 288 pixels, common interface format, CIF) consists of 12 GOBs, each containing 33 macroblocks. A macroblock contains four luminance blocks (block = 8 × 8 pixels) and two chrominance blocks. The simulated system was an 802.3 LAN driven by multiple video sources. Each data packet consisted of one or more GOBs.

They observed that the bit-rate data, at the GOB level, contained a significant periodic component. The reason for this is that each GOB retains significant correlations between the same blocks in successive frames since a block encodes the same portion of a frame and scene content changes slowly over time (seconds). They removed this component from the sample data, and then applied $TES$ to the residual process, $R_n$. The periodic component was determined by using periodogram analysis to estimate the parameters $K$, $\omega_i$, $A_{\mathbf{i}}$ and $B_i$. These parameters are used to model the periodic component as

$$P_n = \sum_{i=1}^{K} (A_i \cos \omega_i n + B_i \sin \omega_i n). \tag{31}$$

The residual process is then,

$$R_n = X_n - P_n. \tag{32}$$

A single-step innovation density with $\alpha = 0.50$ and $\phi = 0.30$ was determined for $R_n$ (using the $TEStool$ software) and the stitching parameter, $\xi$, was set to 0.5. The periodic process was added to the TES process, yielding the final foreground process

$$X_n = H_Y^{-1}(S_{0.5}(U_n^+)) + P_n. \tag{33}$$

The parameters for the model are then $\{\alpha, \phi, \xi, A_i, B_i\}$, where $i = 1, 2 \ldots K$. The model produced an autocorrelation which matched its empirical data counterpart up to a lag of 100 frames. It then was compared to an earlier AR model which did not extract the periodic component, but distributed the bits within a frame equally across GOBs. Simulation showed the TES model produced lower packet loss and delay given the same throughput when compared to results using the AR model.

### *4.3 Frame and slice layer models using a generalized TES (G-TES) process*

Lazar et al. [27] developed both a frame and slice layer models for a DCT-encoded version of the movie *Star Wars*. They used a generalized $TES$ process in which the innovation process is not i.i.d. Each scene was modeled as a stationary process and scene lengths followed a geometric distribution with parameter

$$p = \frac{1}{1 + E[L_n]}, \tag{34}$$

where $E[L_n]$ is the expected value of the duration of a scene, $L_n$. Scene change boundaries were determined using the absolute difference in bit rates of adjacent frames. The model attempts to capture the large change in bit-rate magnitude observed at scene change boundaries. The scene change process was incorporated into the innovation process as

$$V_n = (1-W_n)(L + (R-L)Z_n) + W_n \left(-\frac{\alpha_c}{2} + \alpha_c Z_n\right), \tag{35}$$

where $\{Z_n\}$ is a sequence of i.i.d. random variables with uniform marginals in the interval $[0, 1)$, and $\{W_n\}$ is a sequence of i.i.d. Bernoulli random variables signaling that a scene change occurred. The $GTES$ parameters were determined heuristically to be $\alpha_c = 0.28$ and $R = -L = 0.001$. The average scene length, $E[L_n]$, was found to be about 100 frames.

The same innovation process, with only a slight modification, was used for the slice-layer model. In order to replicate the pseudo-periodic behavior of the autocorrelation function, they added a modulating function $a_n$. The background process was then defined as

$$U_n = \langle U_{n-1} + a_n V_n \rangle, \tag{36}$$

where

$$a_n = \begin{cases} 1 & \text{if } 0 \leq n\%s \leq \dfrac{s}{2} - 1, \\ -1 & \text{if } \dfrac{s}{2} \leq n\%s \leq s - 1, \end{cases} \tag{37}$$

with $s$ being the number of slices per frame and % is the modulo operator.

The value of $E[L_n]$ must now be expressed in terms of slices, so a scale transformation changed it from 100 frames to 3000 slices (30 slices per frame). The parameters $L$ and $R$ were set to 0.003 and 0.008, respectively, while $\alpha_c$ was left intact.

The final model for both the frame and slice layer was

$$X_n = H_Y^{-1}(S_\xi(U_n)), \tag{38}$$

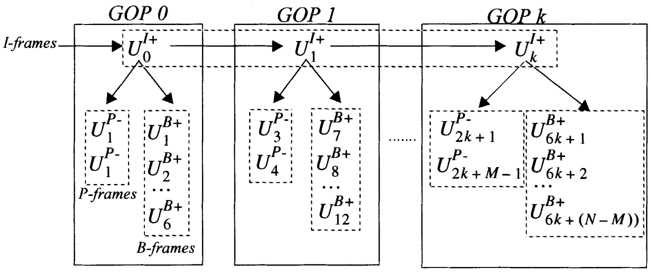


**Fig. 4.** Generation of background processes for $I$, $P$ and $B$ frames in relation to the GOP sequence

where $a_n = 1$ for the frame layer model, and $a_n$ is given by Eq. 37 for the slice-layer model.

The parameters for the model are $\{\alpha_s, \alpha_c, \xi, p\}$ for the frame layer and $\{\alpha_s, \alpha_c, \xi, p, a_n\}$ for the slice layer. The frame-layer model matched the autocorrelation function of the empirical data well up to a lag of about 300 frames, but thereafter it dropped below its empirical counterpart. The slice-layer model matched the autocorrelation well up to a lag of 100 frames. The pseudo-periodic behavior was captured, but the model appeared to underestimate the peaks in the autocorrelation function. The attractiveness of the GTES approach is that it provides a way to model VBR video at two different time scales (slice and frame), while directly incorporating a scene change mechanism. This is the first model considered to be of the hierarchical type. The model was used in simulations to determine the bandwidth allocation required for sources of different service types.

### *4.4 MPEG frame-layer model using a composite-TES (C-TES) processes*

Reininger et al. [28] developed a model for MPEG sequences containing $I$, $B$ and $P$ frames. The model uses the background process $\{U^+\}$ for $I$, $B$ frames and $\{U^-\}$ for $P$ frames, giving the background sequences, $\{U^{I+}\}$, $\{U^{B+}\}$ and $\{U^{P-}\}$. The scene change process was not taken into account in this model.

An interesting point regarding this model is that the number of bits in $B$ and $P$ frames, within a GOP, depends on the $I$ frame located at the beginning of a GOP. Recall that, in MPEG, a GOP consists of a set of $I$, $B$ and $P$ frames arranged in a deterministic pattern which repeats throughout the sequence. For example, given a frame sequence $\{\ldots I_2 B_3 B_4 P_2 B_5 B_6 P_3 B_7 B_8 I_3 \ldots\}$, where the subscript is the index number of the associated frame type, the background sequence variates for $B_3$ and $P_2$ are set equal to the background sequence variate for $I_2$ ($U_2^{I+} = U_3^{B+}) = U_2^{P-}$). This makes intuitive sense, because both $B$ and $P$ frames, within a GOP, reference the $I$ frame in the encoding process. An illustration of the relationship of the background sequences for $I$, $P$ and $B$ frames is shown in Fig. 4.

The final model is composed of a deterministic combination of each frame type process. Process selection is determined by the GOP frame sequence pattern, defined by the MPEG encoding parameters $N$ and $M$, which are the $I$-frame and $P$-frame distances, respectively. The three random processes are defined as

$$X_n^{I+} = H_{Y_I}^{-1}(S_{\xi_I}(U_n^{I+})),$$

$$X_n^{B+} = H_{Y_B}^{-1}(S_{\xi_B}(U_n^{B+})), \qquad (39)$$
$$X_n^{P-} = H_{Y_P}^{-1}(S_{\xi_P}(U_n^{P-})).$$

The model requires nine parameters, $\{\alpha, \phi, \xi\}_{I,B,P}$, three for each frame type. One point of interest here is that MPEG sequences with $IBP$ frames exhibited a pseudo-periodic autocorrelation function similar to that produced by slice-layer sequences. This behavior is caused by the deterministic sequencing of $IBP$ frames and is captured well by this model.

### *4.5 Frame-layer model using a Markov-modulated TES (MRMT) process*

Melamed and Pendarakis [29] developed a model for a DCT-encoded version of the movie *Star Wars* (this is the same sequence used in Sect. 4.3) and take scene changes into account. The approach taken here is different from that in Sect. 4.3 in that the scene change process is not assumed to be i.i.d., but Markovian. The first task was then to segment the video sequence into individual scene segments and classify each segment. Scene boundaries were detected by measuring the sustained absolute difference of the bit rates between a series of successive frames, again, similar to the technique used in Sect. 4.3[10].

Once the video sequence is segmented, each segment is classified using a clustering algorithm based on the minimum Euclidean distance between mean bit rates. Four clusters were sufficient to categorize the video sequence into four scene classes. A $TES$ model was then created for each scene class, where each class was mapped to a state of a Markov chain, illustrated in Fig. 5, whence the name *Markov renewal modulated TES* ($MRMT$) process. Scene durations were class dependent; however, its autocorrelation function was practically zero for lags greater than five frames, so it was assumed to be i.i.d. random variable, represented by $\{T_i\}$, where $i = 1, 2, 3, 4$ is the class index. Although it is not explicitly stated that the scene duration distribution is geometric, the video sequence is the same as that used in [27] where its form is stated explicitly.

The final model consists of four TES processes, one for each class,

$$X_n^{i+} = H_{Y_i}^{-1}(S_{\xi_i}(U_n^{i+})), \quad i = 1, 2, 3, 4 \qquad (40)$$

as well as four renewal processes for the corresponding scene durations.

The parameters for the model are $\{\alpha_i, \phi_i, \xi_i, P\}$, where $i$ is the class index and $P = [p_{i,j}]$ transition probability matrix. This yields 12 parameters for the matrix, and 3 for each TES process giving a total of 24 parameters. The model was used to generate 171,000 samples and the attendant sample path, histogram, autocorrelation function and spectral density were compared to their empirical counterparts. In all cases, excellent matches were achieved. This model produced better matches to the autocorrelation function at long lags, since the Markov chain captures the longer term scene change behavior.

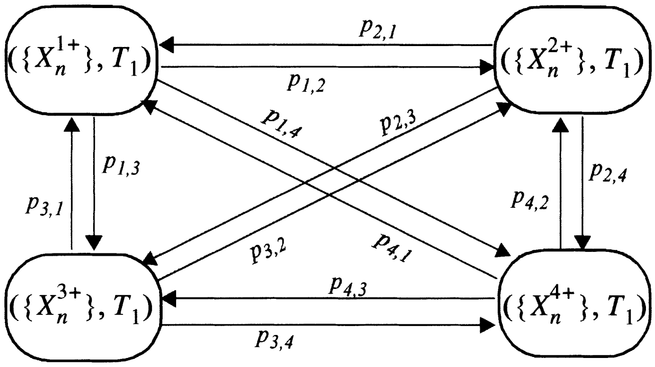


**Fig. 5.** The modulating Markov chain which controls TES process selection and scene duration based on class type. Transition probabilities, $p_{ij}$, from class $i$ to class $j$ are determined from the empirical data of scene transitions. The amount of time spent in a particular state, $T_i$, is sampled from the empirical distribution of class $i$ scene durations. $\{X_n^{i+}\}$ represents the TES process for scene class $i$

## 5 Models based on self-similar processes

Loosely speaking, a process is said to be self-similar if the samples for that process appears "similar", regardless of the duration of the sampling interval (time scale). One of the important characteristics of a second-order self-similar process is that it is also long-range dependent (LRD). An LRD process is defined as a process in which its autocorrelation function is not summable. This behavior differs from short-range-dependent (SRD) processes, whose autocorrelation functions are summable and power spectrums are bounded at low frequencies. Recent work by Beran et al. [30] has shown that long-range dependence is intrinsic to VBR sequences, and given that there is some evidence that LRD processes can negatively affect multiplexing performance [31, 32], source models which capture this characteristic were developed. We present a brief overview of LRD; however, for more information the reader should see [30].

### *5.1 Overview of long-range dependence*

It has been found that LRD processes occur quite often in nature. Natural phenomena such as rainfall, the annual growth of tree rings, river levels and discharges are often described as self-similar processes. Hurst first discovered this property by investigating the amount of storage required in the Great Lakes of the Nile river basin [33]. He found that the expected value of the quantity $R(n)/S(n)$ (*rescaled adjusted range statistic R/S*), asymptotically followed a power law

$$E[R(n)/S(n)] \approx cn^H, \quad n \to \infty \qquad (41)$$

where $c$ is a positive constant independent of $n$, $R(n)$ is the adjusted range, $S(n)$ is the sample standard deviation, and $H$ is the *Hurst parameter* with range $0.5 < H < 1$. The rescaled adjusted range is calculated using

$$\frac{R(n)}{S(n)} \qquad (42)$$
$$= \frac{max(0, W_1, W_2, \ldots W_n - min(0, 0, W_1, W_2, \ldots W_n)}{s(n)},$$

where

[10] It should be noted that sequences containing $I$, $B$ and $P$ frames will require different methods to determine scene change boundaries. It seems feasible that, in this case, this technique could be applied to $I$ frames only.

**Table 3.** Hurst parameters of various traffic types.

| Traffic type | H |
|---|---|
| Computer traffic [34] | $\approx 1$ |
| CNN [35] | 0.90 |
| *Star Wars* (Motion JPEG) [35] | 0.88 |
| *Star Wars* (MPEG IBP) [35] | 0.86 |
| *Star Wars* (B/W, DCT) [36] | 0.83 |
| Video conference [30] | 0.6-0.75 |
| i.i.d [37, 38] | 0.5 |

$$W_0 = 0,$$
$$W_k = \sum_{i=1}^{n} X_i - k\overline{X}(n) \quad k = 1, 2, \ldots, n \tag{43}$$

and $X_i$ is the empirical sequence.

The value $E[R(n)/S(n)]$ is calculated for different values of $n$ and plotted in a *Pox diagram* where log($E[R(n)/S(n)]$) is plotted on the $y$-axis and log($n$) is plotted on the $x$-axis. Linear regression is then used to estimate the Hurst parameter

$$H = \frac{\log\ E[R(n)/S(n)]}{\log\ n}. \tag{44}$$

Typical values of $H$ for various sequences are given in Table 3. Note that the value of $H$ for i.i.d random processes is 0.5, while for computer traffic it is approximately 1. Figure 6 shows the effect of the value of $H$ on a random sequence. We plotted samples generated using a fractional ARIMA process for four different values of H. We see that as $H$ increases, a noticeable low-frequency oscillation is evident in the sequence envelope.

Once $H$ is estimated, a processes such as *fractional ARIMA*, or *fast fractional Gaussian noise* ($ffGn$) is used to create a background sequence, which may be used in turn to generate the foreground sequence using the desired empirical marginal bit-rate distribution. Other methods can be used to estimate $H$ in addition to R/S analysis such as *variance-time* and *periodogram analysis*.

### *5.2 Frame layer model using a fractional ARIMA (f-ARIMA) process*

Garrett and Willinger [36] developed a model a DCT encoding of the movie *Star Wars* (same one used in Sect. 4.3). To match both the left and right tails of the empirical distribution they used a hybrid distribution, $F_{\Gamma/P}$, which consisted of a concatenation of a Gamma distribution for the left tail and a Pareto distribution for the right tail. Right tail matching is particularly important because it describes the probabilities of high bit rates, which can significantly affect queuing performance.

The model uses a fractional $ARIMA(0, d, 0)$ process to generate the background sequence, where $d = H - 0.5$, and its autocorrelation function is given by

$$\rho_k = \frac{d(1 + d) \ldots (k - 1 + d)}{(1 - d)(2 - d) \ldots (k - d)}. \tag{45}$$

The background process, $\{Uk\}$, is generated using Hosking's algorithm[11] [39] which is an $o(n^2)$ algorithm. Each $U_k$ is Gaussian with mean $\mu_k$ and variance $\sigma_k^2$, which are given by the function $\phi$, defined recursively by

$$\phi_{kj} = \phi_{k-1,j} - \phi_{kk}\phi_{k-1,k-j}, \quad j = 1, \ldots, k - 1 \tag{46}$$
$$\phi_{kk} = (N_k/D_k), \tag{47}$$

where

$$N_k = \rho_k - \sum_{j=1}^{k-1} \phi_{k-1,j}\rho_{k-j}, \tag{48}$$

$$D_k = D_{k-1} - (N_{k-1}^2/D_{k-1}), \tag{49}$$

and the initial values are $N_0 = 0$ and $D_0 = 1$. The mean and variance is then

$$\sigma_k^2 = (1 - \phi_{kk}^2)\sigma_{k-1}^2, \tag{50}$$

$$\mu_k = \sum_{j=1}^{k} \phi_{kj} X_{k-j}, \tag{51}$$

where the random variable, $X_0$, is sampled from a standard normal distribution. Finally, the number of bits per frame is represented by the foreground sequence, $\{Y_k\}$, which is generated using the transformation

$$Y_k = F_{\Gamma/P}^{-1}(F_n(U_k)), \quad k > 0 \tag{52}$$

where $F_N$ is the standard normal cumulative distribution function and $F_{\Gamma/P}^{-1}(.)$ is the inverse of the aforementioned hybrid Gamma/Pareto cumulative distribution function. Note that the LRD property is still preserved when transforming from $U_k$ to $X_k$.

The parameters required by the model are ($\mu_\Gamma, \sigma_\Gamma, m_T, H$), where $\mu_\Gamma$ and $\sigma_\Gamma$ are the mean and variance of the Gamma distribution and $m_T$ is the slope of the line which best fits the tail of the Pareto distribution. The model parsimoniously captures the LRD aspect in this sequence using a single parameter.

### *5.3 Frame-layer model using a fast fractional Gaussian noise (ffGN) approximation*

Enssle [40] developed a VBR model for an MPEG version of *Star Wars* containing $I$, $B$ and $P$ frames. The model used the $ffGN$ algorithm to generate the background sequence. This method is an approximation to the fractional Gaussian noise process and has a computational complexity of $o(n)$. Consequently, $ffGN$ provides a faster way to generate the background sequence as compared to *fractional ARIMA*.

The background process, $\{U_s(t)\}$, was generated using $ffGN$. The Hurst parameter, $H$, was estimated to be 0.856, using *periodogram* and *R/S analysis*, which compares well with the value 0.83 found in [36]. The $ffGN$ algorithm requires two additional parameters besides $H$, called the base, $B$, and quality, $Q$. Selecting a value for $B$ in the range $1.1 < B < 2$ and $Q = 20$ produced acceptable estimates

[11] One interesting feature of this algorithm is that, given a discrete autocorrelation sequence, $\rho_k$, a sampla path can be generated whose autocorrelation matches it. We will see this technique used later in a model proposed by Huang et al. [41]

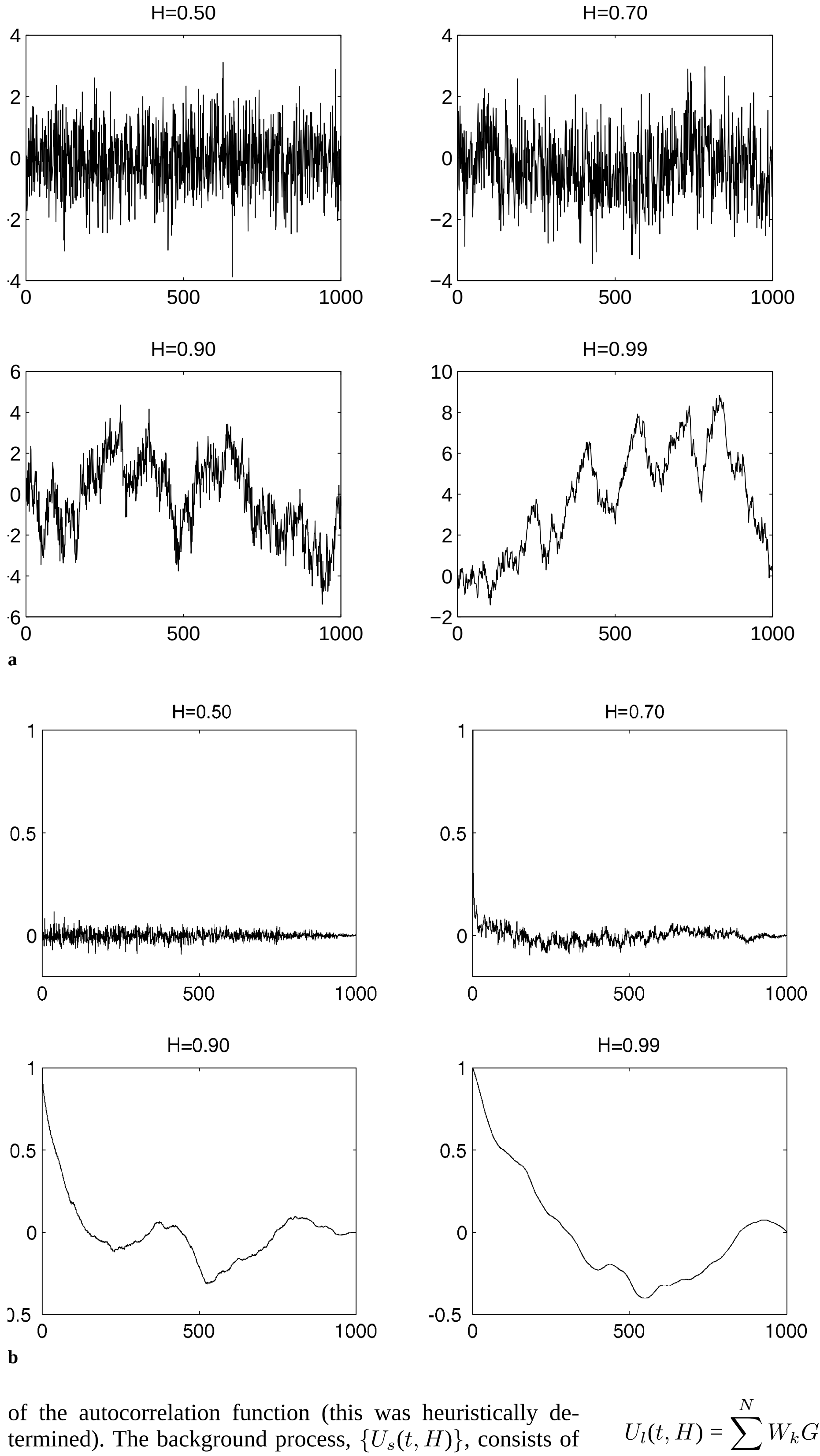


**Fig. 6a,b.** Comparison of the Hurst effect on random process **a** samples generated using fractional ARIMA process (x-axis = sample, y-axis = magnitude) **b** corresponding empirical autocorrelation functions (x-axis = correlation coefficient, y-axis = lag)

of the autocorrelation function (this was heuristically determined). The background process, $\{U_s(t,H)\}$, consists of high- and low-frequency components,

$$U_s(t,H) = U_l(t,H) + U_h(t,H). \tag{53}$$

The low-frequency component is a weighted sum of independent Markov-Gauss processes,

$$U_l(t,H) = \sum_{k=1}^{N} W_k G(t, r(k)). \tag{54}$$

The number of Markov-Gauss processes is

$$N = \lceil \ln(Qn) / \ln(B) \rceil, \tag{55}$$

where $n$ is the length of the time series and the weight factors are

$$W_k = \sqrt{\frac{H(2H-1)(B^{1-H} - B^{H-1})(B^{-2k(1-H)})}{\Gamma(3-2H)}}. \tag{56}$$

The lag-1 covariance is given by

$$r(k) = e^{-B^{-k}}, \tag{57}$$

which is used in the Markov-Gauss process

$$G(t, r(k)) = \begin{cases} G_k(1), & t = 1 \\ r(k)G(t-1, r(k)) + \sqrt{(1 - r^2(k))}G_k(t), & t > 1 \end{cases} \tag{58}$$

where $G_k(t)$ is standard normal.

The high-frequency component of Eq. 53 is

$$U_h(t, H) = \sqrt{1 - \frac{B^{H-1}}{4(1-H)\Gamma(-2H)}}G(t), \tag{59}$$

where $G(t)$ is standard normal.

Each frame type has an corresponding distribution for the number of bits per frame, which was determined to fit the lognormal distribution well. The background process, $U_s(t, H)$, is then distorted using the transformation of the frame type distribution, $F_i^{-1}$, to give

$$X_i(t) = F_i^{-1}(U_s(t)), \quad i = 1, 2, \ldots, n \tag{60}$$

which translates to the following

$$X_i(t) = \exp\left[\sqrt{\ln(1+\Phi)}X_s(t) + \ln E[X_i] - \frac{1}{2}\ln(1+\Phi)\right], \tag{61}$$

where $\Phi = (Var[X_i])/E[X_i]^2$ and $i$ is the frame type $I$, $B$ and $P$.

The model requires three parameters: $B$, $Q$ and $H$ for the background process and six parameters: $E[X_i]$, $Var[X_i]$, for the lognormal distributions, for a total of nine parameters. The model basically operates as follows. At time $t$, the frame type $i$, which is predetermined by the MPEG GOP pattern (IBBPBBP...), is used to select one of the three transform functions: $X_I(t)$, $X_B(t)$ and $X_P(t)$. A sample is then generated of $\{U_s(t, H)\}$ which is used in the selected transform function. Time is then incremented to determine the next frame type and the process continues for the duration of the video sequence.

### *5.4 Frame-layer SRD/LRD model using an f-ARIMA process*

Huang et al. [41] developed a model for an MPEG encoding of the movie *Last Action Hero*, which contains $I$, $B$ and $P$ frames. It is referred to as a unified model because it uses a hybrid function in order to match the autocorrelation function at both the short-term and long-term lags. The Hurst parameter, $H$, was used to generate the LRD part of the autocorrelation function, and a weighted sum of exponentials was used to match the SRD part. These parts were then combined, yielding the hybrid autocorrelation function

$$\rho_k = \begin{cases} e^{-0.00565\,k} & , \quad k < K_t \\ 1.59\,k^{-0.2} & , \quad k \geq K_t \end{cases} \tag{62}$$

where $K_t$ is the boundary between the SRD and LRD parts. The Hurst parameter was determined using both R/S and variance-time analysis and the SRD parameters were determined by using regression analysis. Once the autocorrelation function was determined, Hosking's algorithm was used to define the background process, $\{U_n\}$. Individual histograms were constructed for each frame type and the attendant background process was then transformed for each frame type via

$$X_n^i = H_Y^{-1}(U_n). \quad i = I, B, P \tag{63}$$

Histograms were used since the authors determined that the empirical distributions did not seem to fit any of the existing classical distributions.

The model requires four parameters to estimate $\rho_k$, the Hurst parameter $H$, and one parameter for each cell in the frame type histogram. Samples were produced which matched the autocorrelation function well for lags up to 500 frames. The portion of the autocorrelation function up to lag $K_t \sim 75$ is fitted well by the exponential function. This is probably the most accurate model to date in terms of matching the autocorrelation function for both short and long lags.

## 6 Analytical and non-Markovian type models

The analytical model proposed by Marafih et al. [42] is a model whose premise relies on a discrete-state Markov chain and uses analytical methods, such as stationary-interval (SI), asymptotic method (ASM) and queuing network analyzer (QNA), in order to estimate multiplexer cell loss. Model parameters were estimated using the first two moments of the empirical sequence ($I$ and $P$ frames). Results showed that the SI model produced results which provided a good upper bound, while the ASM model provided a lower bound. The QNA model seemed to match the results using the empirical data.

Skelly et al. [43] proposed a generalized histogram-based analytical model for ATM VBR sources. Each source is characterized by a bit-rate histogram and the histogram of the aggregate (multiplexed) process is obtained by convolving the individual source histograms. Once obtained, the histogram is used to solve an M/D/1/K system and the buffer occupancy distribution is given by

$$P(n) = \sum_{I=1}^{N} P(n|\lambda_I = \lambda)P(\lambda = \lambda_I), \tag{64}$$

where $P(\lambda = \lambda_I)$ is the bit-rate histogram approximation of $f_\lambda(x)$, $P(n|\lambda = \lambda_I)$ is the buffer occupancy distribution given the arrival rate is $\lambda_I$, and $N$ is the number of bins in the histogram. In order to generalize the applicability of their model, they used an MMPP/$E_k$/1/K as the source model to determine the buffer occupancy distributions. They found that a Markov chain of eight states was sufficient; thus, requiring an 8 × 8 transition probability matrix. This model requires 64 parameters for the matrix and 8 parameters for the average arrival rate for the Poisson process in each state, for a total of 72 parameters.

Krunz et al. [44] proposed a model for an MPEG encoding of the movie *The Wizard of Oz* consisting of 41,760 frames ($I$, $B$ and $P$). Each frame type was fitted using a lognormal distribution, where parameters were estimated using the maximum likelihood estimators (MLE). The frame size process for each frame type was assumed to be i.i.d., which greatly simplifies the model. Selection of the appropriate frame type processes was done deterministically based on the MPEG GOP structure. While this model is simplistic, simulation results for cell loss, when multiplexing up to 10 sources, matched the results produced by the empirical sequence fairly well. A similar model was proposed by Enssle [35] for a full-length (123,574 frames) MPEG encoding of the movie *Star Wars*.

## 7 Summary

It is apparent that one has many choices when it comes to VBR models. In order to narrow the selection, it is important that the user define the attributes of the source in order to assist in model selection. Some questions to answer are

1. does the VBR sequence contain significant scene changes?
2. What is the frame type sequence pattern? Is it H.261 or MPEG?
3. Does the sequence contain $B$ frames?
4. Does the frame type distribution fit any of the classical distributions, such as lognormal, Gamma, etc.?
5. What level is to be modeled? Frame, slice, GOB, or ATM cell?

Answering these questions will assist the user in determining the proper VBR model to use in simulations. A sequence which contains different frame types $(I, B, P)$ would likely require a model which uses separate processes for each frame type, so this is an important consideration. Another matter to consider is the marginal distribution of the source process. If a classical distribution will fit, then samples can be generated using many of the existing mathematical transformations; otherwise, samples must be generated using the inversion method on the empirical histogram. We summarize the attributes of the models covered in this survey in Table 4.

**Table 4.** Summary of VBR model attributes

| # | Model type | Parameters | Scene change | Level | $B$ frames? |
|---|---|---|---|---|---|
| 1 | ARMA [14] | 1003 | No | subframe | No |
| 2 | DAR[a] [15] | 4 | No | Frame | No |
| 3 | MRP [16] | 40 | No | Frame | No |
| 4 | AR/MC [17] | 8 | Yes | Frame | No |
| 5 | MC [18] | 51–102 | No | Frame/Slice | No |
| 6 | AR/MC [19] | 9 | No | Frame | No |
| 7 | AR [20] | 20 | No | Block | No |
| 8 | DAR/i.i.d [21] | 10 | Yes | Frame | No |
| 9 | DAR/i.i.d. [22] | 5 | Yes | Frame | No |
| 10 | TES [26] | 2K[b]+3+H[c] | No | GOB | No |
| 11 | GTES [27] | 5+H | Yes | Frame/slice | No |
| 12 | composite-TES[28] | 9+3H | No | Frame | Yes |
| 13 | MRMT [29] | 24+4H | Yes | Frame | No |
| 14 | f-ARIMA [36] | 4 | Yes | Frame | No |
| 15 | ffGN [37] | 9 | Yes | Frame | Yes |
| 16 | Unified f-ARIMA [41] | 4+3H | Yes | Frame | Yes |
| 17 | $\Sigma$GI/D/1 [45] | 2 | No | Frame | No |
| 18 | MMPP/$E_k$/1/K [45] | 72 | No | Frame | No |
| 19 | i.i.d [32] [42] | 6 | No | Frame | Yes |

[a] MC = Markov chain

[b] K = óf frequency coefficients to match periodic GOB process

[c] "H" is number of histrogram cells required by a TES process to match the empirical distribution

### 7.1 Model recommendations

As a result of this survey, several recommendations can be given which might be helpful in determining an appropriate model for VBR video. The first is that, if a video conference model is required (no $B$ frames), and there are few or no scene changes, then a DAR model is a better choice than an AR model. Another contender, however, is the MRP model, which produced better results than DAR, but requires more parameters. It is conceivable, that, for certain video sequences in which the bit rate stays at particular levels for long periods of time, MRP would be a better choice.

Another recommendation deals with the modeling of the scene change process. When modeling sequences with numerous scene changes, the user can choose either a Markov-based model or a self-similar model. The choice, however, is not clear cut. On the one hand, self-similar models have the advantage that only a single parameter $(H)$ is required; whereas, a Markov chain requires many parameters (transition probabilities). Unfortunately, the drawback to self-similar models is the computational complexity involved in generating samples can be quite high. This is alleviated somewhat by using $ffGN$. Nevertheless, it does appear from the results given in [36] and [40] that ignoring the scene change process will most likely yield overly optimistic simulation results.

The last recommendation deals with models for MPEG video which contain $I$, $B$ and $P$ frames. It seems clear that these models require separate processes for each frame type. The assumption to select each process deterministically according to the GOP pattern is valid. However, this is due to the fact that the GOP pattern is fixed for the duration of the sequence. Future encoders will most likely vary frame type sequencing based on video content, which will vary the GOP pattern. In fact, this mode of operation is fully compliant with the MPEG standard. For such sequences, the frame type selection process will also be stochastic, thereby invalidating the deterministic assumption. This opens a new area which is yet to be covered in VBR modeling.

### 7.2 Self-similar models

One issue concerning self-similar models is the relative importance in matching the SRD component of the autocorrelation function. We saw that in Huang et al.'s model [41] both the SRD and the LRD components were matched. In effect, the background process exhibits both a rapid decay at short lags and a much slower decay for long lags. Enssle's model (ffGN) does not do this, but claims that the SRD component is sufficiently captured by the deterministic GOP process. A question which arises then is the added benefit derived in capturing the SRD component of an individual frame type. With regard to the question of the relative significance of the

LRD component in VBR sequences, recent work by Heyman and Lakshman [11] suggests that short busy periods (typical when multiplexing VBR sequences) and finite buffers cause a "reset effect" which, for the most part, negates any LRD effects. As a consequence, they argue that SRD, which is modeled well by using Markov processes, is a more important aspect of VBR sequences.

### *7.3 Recent models*

Two new models, which have recently been published during the review of this survey, deserve mention. The first is the model by Krunz and Tripathi [46] for an MPEG VBR sequence containing $I$, $B$ and $P$ frames. The model consists of a composite of three random variables; one for each frame type. The random variables for both $B$- and $P$-frame types was found to be i.i.d. with a lognormal distribution. For $I$ frames, the scene change process, which was both geometric and i.i.d., was used to define an average bit-rate level, while an AR(2) process was used to produce the fluctuations over this level. The GOP pattern was used to select one of the three random variables. The model by Jelenkovic et. al. [47], interestingly, does not model the MPEG sequence at the frame layer, rather it models the sequence at the GOP layer. Scene changes are discerned by taking the normalized second-order difference between GOP sizes and determining if it is both large and negative. Scenes are then grouped into four classes based on a clustering of GOP sizes (similarly done in [29]). Each class was then modeled using an i.i.d. process which was selected using a Markov chain. The paper also addresses the controversy over the relative importance of the SRD and LRD components in an MPEG VBR sequence.

### *7.4 Further research*

Most of the focus in VBR research has centered on packet loss at a multiplexer buffer. This metric is assumed to be directly related to video quality of service (QoS). However, aggregate loss might not be sufficient in determining end-user "perceived" quality. In fact, loss "bursts" and their frequency of occurrence might be a more significant metric. More specifically, VBR streams with similar aggregate loss, but significantly different mean time between losses (MTBL), might be perceived as having very different quality by the end user (this phenomena has been suggested in [36] and [45]). As a consequence, research is needed to determine how useful these models are in predicting this metric.

Another direction for further work is the performance of a model when using a "smoother" mechanism. A smoother reduces the source traffic bit-rate peak-to-mean ratio by injecting extra delay. Models which do not accurately reproduce the empirical sample path might not be suitable when used in conjunction with a smoother. The only paper in this survey which partially addresses this issue is the one by Lucantoni et al. [16], where leaky-bucket contour curves were studied. However, the leaky bucket is more of a traffic policing device than smoothing device and there are a whole class of smoothers which have not been studied in conjunction with any of the proposed models. In addition, the buffer service policy used by the multiplexer should be taken into account. Most of the models assume a first-come-first-serve policy, but others exist such as weighted fair queuing.

Another area often overlooked (or minimized in its significance), is the cross-correlation effect between MPEG sequences which result from the GOP pattern. Sequences which have GOP parameters $(N, M)$ which are the same, or multiples thereof, can have $I$ and $P$ frames which overlap in time. Also, since these frames are much larger than $B$ frames, their overlap could cause serious degradations in multiplexer performance. Since the GOP pattern is deterministic, this overlap will continue for the duration of the video sequence.

Finally, while it appears that a single model will not apply to all video sequences, a generalized modeling framework is required which can be applied to a wide variety of video content. It is not clear that this objective can be achieved using Markovian-based models, since none have been applied to MPEG sequences containing $B$ frames. As to whether a hierarchical model using Markov chains and DAR can be used for such sequences, remains to be seen. The two candidates found in this survey are the self-similar model, proposed by Enssle, and the MRMT model, proposed by Melamed and Pendarakis. Given the arguments by Heyman that the LRD aspect of VBR sequences is relatively unimportant when compared to the SRD aspect, it would appear that a model such as MRMT can provide such a framework.

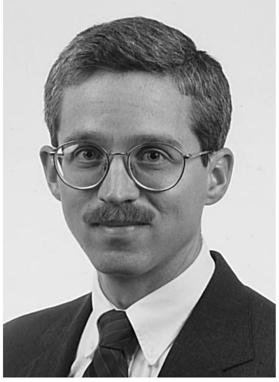
*Acknowledgements.* The authors would like to thank Benjamin Melamed for his insightful comments. They would also like to thank Mark Garrett, Jurgen Enssle and the anonymous reviewers for their helpful comments. This work was supported by the Air Force Office of Scientific Research under Grants F49620-92-J-0441 and F49620-96-1-0061, and by an IBM Corporation Resident Study Fellowship. To contact the authors please send email to mikeiz@mmac.com or reeves@eos.ncsu.edu

## References

1. CCITT Study Group XV (1990) Recommendation of the H-Series. CCITT, Report R31, Geneva Meeting, 16–27 July 1990, pp 79–107
2. Working Party XV/1 Experts Group for ATM Video Coding (1992) IVS Baseline Document. CCITT, Working Party XV/1 Experts Group for ATM Video Coding
3. ISO/IEC (1993) Information Technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s. ISO/IEC 11172. ISO/IEC, Geneva, Switzerland
4. ISO/IEC (1994) Information Technology – Generic Coding of Moving Pictures and Associated Audio Information. ISO/IEC 13818. ISO/IEC, Geneva, Switzerland
5. Haskell B (1972) Buffer and Channel Sharing by Several Interframe Picturephone coders. Bell Syst Tech J 51(1): 261–289
6. Kishino F, Manabe K, Hayashi Y, Yasuda H (1989) VBR Coding of Video Signals for ATM Networks. IEEE J Sel Areas Commun 7(5): 801–806
7. Morrison DG (1990) Variable bit-rate Coding for Asynchronous Transfer Mode Networks. Br Telecommun Tech J 8(3): 70–80
8. Verbiest W, Pinno L, Voeten B (1988) The Impact of the ATM Concept on Video Coding. IEEE J Sel Areas Commun 6(9): 1623–1632
9. Lion M (1991) Overview of the px64 kbits/s Video Coding Standard. Commun ACM 34(4): 60–63
10. Le Gall D (1991) MPEG: A Video Compression Standard for Multimedia Applications. Commun ACM 34(4): 60–63

11. Heyman D, Lakshman TV (1996) What Are the Implications of Long-Range Dependence for VBR-Video Traffic Engineering? IEEE/ACM Trans Networking 4(3): 301–317
12. Proakis JG (1989) Digital Communications, 2nd Edition, McGraw-Hill, New York
13. Maglaris B, Anastassiou D, Sen P, Karlsson G, Robbins JD (1988) Performance Models of Statistical Multiplexing in Packet Video Communications. IEEE Trans Commun 36(7): 834–843
14. Grunenfelder R, Cosmas JP, Manthorpe S, Odinma-Okafor A (1991) Characterization of Video Codecs as Autoregressive Moving Average processes and Related Queuing System Performance. IEEE J Sel Areas Commun 9(3): 284–293
15. Heyman D, Tabatabai A, Lakshman TV (1992) Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks. IEEE J Sel Areas Commun 2(1): 49–59
16. Lucantoni DM, Neuts MF, Reibman A (1994) Methods for Performance Evaluation of VBR Video Traffic Models. IEEE/ACM Trans Networking 2(2): 176–180
17. Ramamurthy G, Sengupta B (1992) Modeling and Analysis of a Variable Bit Rate Multiplexer. IEEE INFOCOM 2: 817–827
18. Pancha P, El Zarki M (1993) Bandwidth-Allocation Schemes for Variable-Bit-Rate MPEG Sources in ATM Networks. IEEE Trans Circuits Syst Video Technol 3(3): 190–198
19. Yegenoglu F, Jabbari B, Zhang YQ (1993) Motion-Classified Autoregressive Modeling of Variable bit-rate Video. IEEE Trans Circuits Syst Video Technol 3(1): 42–53
20. Jabbari B, Yegenoglu F, Kuo Y, Zafar S, Zhang YQ (1993) Statistical Characterization and Block-Based Modeling of Motion-Adaptive Coded Video: IEEE Trans Circuits Syst Video Tech 3(3): 199–207
21. Heyman D, Lakshman TV (1996) Source Models for VBR Broadcast-Video Traffic. Trans Networking 4(1): 40–48
22. Frater MJ, Arnold JF, Tan P (1994) A New Statistical Model for Traffic Generated by VBR Coders for Television on the Broadband ISDN. IEEE Trans Circuits Syst Video Technol 4(6): 521–526
23. Hill JR, Melamed B (1995) TEStool: A visual interactive environment for modeling autocorrelated time series. Performance Eval 24: 3–22
24. Jelenkovic PR, Melamed B (1995) Algorithmic modeling of TES processes. IEEE Trans Autom Control 40(7): 1305–1312
25. Melamed B (1993) An Overview of TES Processes and Modeling Methodology. Performance Evaluation of Computer and Communications Systems, Lecture Notes in Computer Science. Springer, Berlin Heidelberg New York, pp 359–393
26. Melamed B, Raychaudhuri D, Sengupta B, Zdepski J (1994) TES-Based Video Source Modeling for Performance Evaluation of Integrated Networks. IEEE Trans Commun 42(10): 2773–2777
27. Lazar A, Pacifini G, Pendarakis DE (1993) Modeling video Sources for Real-Time Scheduling. IEEE GLOBECOM 2: 835–839
28. Reininger D, Melamed B, Raychaudhuri D (1994) Variable bit-rate MPEG Video: Characteristics, Modeling and Multiplexing. In: Proc. of the 14th Int. Teletraffic Congress (ITC 14), 6–10 June 1994, Antibes Juan-les-Pins, France, pp 295–306
29. Melamed B, Pendarakis D (1994) A TES-Based Model for compressed Star Wars Video. IEEE GLOBECOM, 120–126
30. Beran J, Shreman R, Taqqu MS, Willinger W (1995) Variable-bit-rate video traffic and long range dependence: Part 3. IEEE Trans Commun 43(2–4): 1566–1579
31. Adas A, Mukherjee A (1995) On Resource Management and QoS Guarantees for Long Range Dependent Traffic. IEEE INFOCOM 2: 779–787
32. Likhanov N, Tsybakov B, Georganos ND (1995) Analysis of an ATM Buffer with Self-Similar (“Fractal”) Input Traffic. IEEE INFOCOM 3: 985–992
33. Hurst HE (1951) Long-Term Storage Capacity of Reservoirs. Trans Am Soc Civ Eng 116: 770–799
34. Leland WE, Taqqu MS, Willinger W, Wilson DV (1994) On the Self-Similar Nature of Ethernet Traffic. IEEE/ACM Trans. on Networking, Vol. 2, No. 1, Feb. 1994, pp 1–5
35. Enssle J (1994) Modeling and Statistical Multiplexing of VBR MPEG Compressed Video in ATM Networks. In: Proc. 4th Workshop on High Speed Networks, 7–9 September 1994, Brest, France, pp 59–67
36. Garrett MW, Willinger W (1994) Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. ACM SIGCOMM 269–280
37. Feller W (1951) The Asymptotic Distribution of the Range of Sums of Independent Random Variable. Ann Math Stat 22: 427–432
38. Mandelbrot BB, Ness JW van (1968) Fractional Brownian Motions, Fractional Noises and Applications. SIAM Rev 10: 422–437
39. Hosking JRM (1984) Modeling Persistence in Hydrological Time Series Using Fractional Differencing. Water Resources Res 20(12): 1898–1908
40. Enssle J (1995) Modeling of Short- and Long-Term Properties of VBR MPEG-Compressed Video in ATM Networks. In: Proceedings of the 1995 Silicon Valley Networking Conference & Exposition, April 1995, San Jose, Calif., pp 95–107
41. Huang C, Devetsikiotis M, Lambadaris I, Kaye R (1995) Modeling and Simulation of Self-Similar Variable bit-rate Compressed Video: A Unified Approach. ACM SIGCOMM 114–125
42. Marafih NM, Zhang Y-Q, Pickholtz RL (1994) Modeling and Queueing Analysis of Variable-Bit-Rate-Coded Video Sources in ATM Networks. IEEE Trans Circuits Syst Video Technol 4(2): 121–128
43. Skelly P, Schwartz M, Dixit S (1993) A Histogram-Based Model for Video Traffic Behavior in an ATM Multiplexer. IEEE/ACM Trans Networking 1(4): 446–459
44. Krunz M, Sass R, Hughes H (1995) Study of VBR MPEG-coded video traffic and associated multiplexing performance. Comput Syst Sci Eng 11: 135–143
45. Ramamurthy G, Raychaudhuri D (1995) Performance of Packet Video with Combined Error Recovery and Concealment. In: IEEE Infocom, Vol. 2, pp 153–161
46. Krunz M, Tripathi SK (1997) On the Characterization of VBR MPEG Streams. In: ACM Sigmetrics, 1997, Seattle, Wash. pp 192–202
47. Jelenkovic PR, Lazar AA, Semret N (1997) The Effect of Multiple Time Scales and Subexponentiality in MPEG Video Streams on Queueing Behavior. IEEE J Sel Areas Commun 15(6): 1052–1071

Michael R. Izquierdo is currently a Ph.D. student at North Carolina State University. He received his BSEE from the University of Florida in 1982 and MSEE from Florida Atlantic University in 1987. He has been employed with IBM in Research Triangle Park, North Carolina in the Network Hardware Division as an Advisory Engineer and is currently employed with Multimedia Access Corp., Osprey Technologies as a Senior Staff Engineer. His research interests include video encoding and modeling, packet video and video multiplexing.



Douglas S. Reeves received his Ph.D. in Computer Science from Pennsylvania State University in 1987. Since that time, he has been a member of the faculty at N. C. State University, where he is currently Associate Professor of Computer Science and Electrical and Computer Engineering. His research interests include real-time communication, multimedia, and parallel computing.