J. King Saud Univ., Vol. 18, *Eng. Sci*. (1), pp. 19-56, Riyadh (1426H./2005)

Regression Video Traffic Models in Broadband Networks

A. Alheraish, S. Alshebeili and T. Alamri
Department of Electrical Engineering, College of Engineering,
King Saud University, P.O. Box 800, Riyadh 11421,
Saudi Arabia

(Received on 26 October, 2003; accepted for publication 18 December, 2004)

Abstract. Recently, there has been significant progress towards the deployment of broadband integrated services digital networks capable of flexibly supporting digital video technologies. Digital video such as video on demand, video teleconferencing, video telephony, and broadcast HDTV will constitute a major traffic component of these networks. Statistical analysis and performance modeling of various types of video traffic are required to estimate network resources and predict the behavior of network under various conditions. For over a decade, there has been an enormous amount of interest and research in traffic modeling of compressed variable-bit-rate (VBR) video. An important class of video models that has received much attention lately is regression models. This paper presents a survey of state of the art of regression traffic studies that have been proposed in the literature to model a variety of video applications. The video models will be classified into two classes: teleconference video, and full motion video. In each class, we will highlight the main features, describe the underlying model, and examine the advantages and limitations. Where appropriate, we also show some results of the statistical properties and the ability of the model to predict the queuing performance of a single and multiplexed video sequence.

1. Introduction

Broadband networks such as ATM (Asynchronous Transfer Mode) is capable of handling a wide mixture of traffic sources ranging from data, voice, image, and video. Among these video is expected to be a significant source of network traffic generated by emerging multimedia applications. Video signal can be encoded and compressed as Constant Bit Rate (CBR) or Variable Bit Rate (VBR). CBR codecs keep the output rate constant by using large buffers to regulate bit rate variations. Allocating network resources for CBR video transmission is relatively easy, since the bit rate does not change. However, CBR suffers from disadvantages such as variable video quality, end-to-end delay, and relatively high transmission cost.

For VBR video, the data rate is allowed to vary over time while the picture quality potentially remains constant. VBR provides better video services and a framework to achieve higher resource utilization by exploiting statistical multiplexing techniques. Therefore, VBR video is more efficient than CBR when transported over broadband networks. On the other hand, VBR video source generates traffic with complex characteristics arising from: 1) the use of video compression techniques such as DPCM, H.261, JPEG, MPEG, subband coding, and wavelet; 2) the rapid movement of video activity level and abrupt scene changes; 3) the type of video applications such as video teleconference, video phone, video on demand, and full motion video. This presents a greater challenge in the design of communications, transmission networks, and the associated traffic control such as call admission control, usage parameter control, buffer allocation, and congestion control.

The knowledge of VBR traffic characteristics and the accurate study of its effect on the network performance play a central role in the network design. A direct method to achieve this aim is to perform a live experiment using real networks and real sources. However, testing real networks is quite expensive and often difficult to generate reasonable results. Testing a real video sequence is possible, but the availability of these sequences is still limited. An alternative to this is to model the traffic using a set of rules (mathematical analysis or simulation) that govern the generation bits, cells, or packets within a particular traffic sequence. One advantage of source modeling is that they can be used reasonably quickly to estimate network resources and predict the behavior of network under various conditions, so that network performance can be optimized without requiring the actual video traces. Traffic models are stochastic in nature, and hence many realizations that represent the actual data can be obtained by varying model parameters. Simulation-based performance evaluation of communication protocols concerning video traffic is a flexible tool used widely today. Its flexibility stems from the fact it consists of a computer program that behaves like the traffic under study. Unlike analytical models, which often require many assumptions and are too restrictive for most real-world traffic, simulation modeling places few restrictions on the classes of traffic under study. For communication networks, developing a simulation program requires: modeling random user demands for network resources, characterizing network resources, and estimating system performance based on output data generated by the simulation

Broadly speaking, a good video model can be evaluated by four criteria. First, the model should match certain statistical characteristics of a real video sequence, namely: probability density function, mean, variance, peak, autocorrelation (or possibly higher order statistics), and coefficient of variation of the bit rate. Second, the synthetic video sequence should be similar to real video sequence, so that it can be used to predict a desired performance metric, which may include but not limited to cell loss probability, delay, buffer size, statistical multiplexing gain and bandwidth. Third, the model should be simple and able to generate synthetic video sequence with low computational

complexity. Fourth, the model should characterize a wide range of video sources ranging from low, medium to high motion activity.

In the past decade, traffic modeling has been an active research area. Several good models for VBR video traffic have been proposed in literature. These models can be broadly classified into four categories:

1. Markov models consisting of a Markov process regulating an arbitrary rate process including Markov Modulated fluid models.
2. Long range dependent models (or self-similar models) that characterize network traffic at multiple time scale.
3. Regression models that define the next random variable in the sequence as an explicit function of previous ones within a time window.
4. Hybrid models that combine both Markov and Regression processes.

Among these models, the focus of this paper will be on regression models. In the course of our studies, we have found a need for an exclusive coverage of regression models for several reasons: 1) They are more familiar models and easier computationally; 2) they are versatile stochastic processes for modeling autocorrelated compressed VBR video (i.e. bursty) models; and 3) a lot of new regression models have appeared in the recent studies which need to be thoroughly investigated.

In this paper, we survey a number of regression models. Some of the models are more appropriate than others for a given type of application. Therefore, it is reasonable to classify the video models into two classes: teleconference video and full motion video. The first class consists of video scenes in which one or several people are talking with very little movement and almost unchanged background. The regression models that have been proposed for such class are: simple autoregressive (AR) process, discrete autoregressive (DAR), gamma beta autoregressive (GBAR), gamma autoregressive (GAR), continuous DAR (C-DAR) and general (AR). Full motion video consists of video scenes with rapid movement and frequent scene changes as in general TV. For this class, we focus on the following models: Motion classified AR, composite AR, scene based models, papered autoregressive (PAR), GOP GBAR model, nested AR, and NAR model. In our survey presentation, we use the following methodology for each underlying regression model:

1. We give a short description and highlight the main features.
2. We examine advantages and limitations.
3. Where appropriate, we show some results of the statistical properties and the ability of the model to predict the queuing performance of a single and multiplexed video sequence.
4. We point out the validation procedure.

A number of survey papers have previously been published in the area of VBR video modeling [1-4]. The focus of these papers was on general aspects of AR, Markov,

self-similar, and hybrid models. None of these studies discussed a variety of techniques that have been proposed for regression models. Unlike our work, they consider only a simple AR process for videoconference type.

The rest of this paper is organized as follows. Section 2 defines the AR process. Section 3 and Section 4 examine the video conference models and full motion video models respectively. Summary, recommendation and open issues are discussed in Section 5.

2. Basic Definition of AR Process

Consider a linear system with input *e(n)* and output *x(n)*, where *n* is the discrete time. The finite AR process is generally expressed as:

$$x(n) = \sum_{k=1}^{p} a_k x(n-k) + e(n) \tag{1}$$

where *e(n)* is an uncorrelated process with zero mean and variance σ^2, and $\{a_k, 1 \le k \le p\}$ is a finite sequence with $a_p \neq 0$. Such a process is denoted by *AR(p)* and *p* is called the order of the *AR* process. The sequence {e(n)} consists of *i.i.d* random variables, known as the residual (or error process), that gives the AR model its stochastic nature. The residuals are often normally distributed, which implies that x(n) is also normally distributed, but with different mean and variance. There are a number of methods to estimate the parameters for an AR process given x(n). In the present study, the linear prediction method will be described next.

Given a set of past samples of the signal x(n), a linear approximation of the present value of the signal is defined as:

$$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{2}$$

By subtracting $\hat{x}(n)$ from the current signal value x(n), we obtain the residual signal e(n); that is,

$$e(n) = x(n) - \hat{x}(n)$$
$$= x(n) - \sum_{k=1}^{p} a_k x(n-k) \tag{3}$$

Let

$$\breve{\mathrm{x}} = [\mathrm{x(n)} - \mathrm{x(n-1)} \ldots - \mathrm{x(n-p)}]^{\mathrm{T}}$$
$$\mathring{\mathrm{a}} = [\mathrm{a}_0 \ -\mathrm{a}_1 \ -\mathrm{a}_2 \ \ldots -\mathrm{a}_{\mathrm{p}}]^{\mathrm{T}} \tag{4}$$

where $a_0 = 1$. Then Eq. (3) can be written in the following form:

$$\mathrm{e(n)} = \breve{\mathrm{x}}^{\mathrm{T}} \breve{\mathrm{a}} \tag{5}$$

The most commonly used method to optimize model coefficients $\{a_k\}$ is to minimize the mean-square value ε of the error sequence *e(n)*. Based on (5), we can write:

$$\varepsilon = \mathrm{E}\{\mathrm{e}^2(\mathrm{n})\}$$
$$= \mathring{\mathrm{a}}^{\mathrm{T}} \mathrm{R} \mathring{\mathrm{a}} \tag{6}$$

where *E{ . }* is the expectation operation, and R is the *(p+1)* x *(p+1)* correlation matrix of the input vector x. The prediction model vector a is therefore chosen so as to minimize ε subject to the constraint that the first element of a equals 1. The solution of this constrained minimization problem is well known and is given by [19]

$$\mathring{\mathrm{a}} = \frac{\mathrm{R}^{-1} \breve{\mathrm{a}}_{\circ}}{\delta \mathrm{R}^{-1} \delta} \tag{7}$$

where δ is an (p+1) x 1 vector of the form

$$\delta = [1\ 0\ 0\ ...\ 0]^T \tag{8}$$

For a stationary process with mean μ and variance σ^2, the autocorrelation function (ACF) of x(n) is defined as

$$\rho_k = \frac{E[(x(n)-\mu)(x(n+k)-\mu)]}{\sigma^2} \tag{9}$$

The ACF of an AR(p) can be written as a difference equation:

$$\rho_k = \sum_{r=1}^{p} a_r \rho_{k-r} \tag{10}$$

where ρ_k is the ACF at lag k.

3. Video Conference Models

Recently, video conferencing have been extensively used in IP, ATM networks, and TV broadcasting as a means of interactive communications. Teleconferencing video traffic consists of video scenes in which one or more people are talking with low to medium motion and almost unchanged background. The scene is defined as a segment of a movie with no abrupt changes, but possible with some panning and zooming. A simple motion-compensated interframe coder will suffice for such applications. To reflect the properties of video signals in the design of communication and transmission networks, modeling this teleconference video has received considerable attention. The basic characteristics of teleconferencing traffic are: bell-shaped distribution, high interframe autocorrelations, and that the form of the autocorrelations is essentially exponential. This makes AR models suitable for modeling this type of video. Since teleconferencing scenes do not exhibit a wide range of motion activities and scene changes occur only rarely, the bit rate (or frame size) distribution may be represented closely by a single AR process. Several good AR models have been proposed in the literature such as: simple AR, DAR, GBAR, GAR, C-DAR, and General AR to appropriately characterize bit rate variation of video conferencing signals. This section will describe these models and presents their merits and limitations.

3.1. Simple AR and DAR models

AR(1) model was one of the earlier models proposed for video conference-like traffic. Maglaris *et al.* [7] used a simple AR(1) model to characterize the frame-size of a 10-second long sequence of video under the conditional replenishment compression algorithm. In their model, the AR process was defined as:

$$x(n) = a_1 x(n-1) + e(n) \tag{11}$$

where x(n) is the bit rate of the coded video during the *n*th frame, e(n) is a Gaussian process with variance σ^2, a_1 is the correlation coefficient among two successive frame rates. This process was found to closely represent the bit-rate density of video scenes with slow varying bit rates. The model has proven to be useful for queuing simulation but not appropriate for queuing analysis due to its mathematical complexity. For example, in analyzing the network cell loss ratio (or packet loss ratio), one wants to evaluate the region of extremely low probability density, preferably with analytic technique. Nomura *et al.* [8] also modeled the video conference as an AR(1) process. Two 20-second video sequences were analyzed. One sequence was an active scene and the other is an inactive scene. They suggested modeling video using multiple AR process where a Markov chain was used to determine which AR process was active. However, first order AR can not accurately model correlated traffic. Heyman *et al.* [9] analyzed a 30-min video sequence of 48,500 frames at the ATM cells per frame level with no scene changes. They proposed that the Gamma distribution fit the cells/frame distribution well. They suggested that possibly a mixture of Gamma and exponential distributions might result in a more accurate distribution. They showed that an AR(2) model process with *i.i.d* residual processing a normal distribution fitted the data well and produced too few cell losses in simulation better than AR(1). However, the model did not accurately model frames with a large number of cells and thus underestimated cell loss probabilities. To overcome this problem, the authors suggested a discrete AR(1) (DAR(1)) process to model scene dynamic in which frame sizes are generated according to a finite-state Markov chain. The transition matrix $\boldsymbol{P}$ of this Markov chain is given by:

$$\boldsymbol{P} = \rho \boldsymbol{I} + (1-\rho)\boldsymbol{Q} \tag{12}$$

where ρ is the autocorrelation coefficient and $\boldsymbol{I}$ is the identity matrix and each row of $\boldsymbol{Q}$ consists of the negative-binomial probabilities. The frame sizes stays constant during a scene but varies from one scene to another according to negative-binomial probabilities. By using equation (12), the parameters of negative-binomial distribution were reduced to mean, peak rate, variance and the first order autocorrelation coefficient. The negative-binomial probabilities are essentially the sum of independent geometric (discrete equivalent of exponential) random variables. This means that the residual process selected for DAR(1) model is non-Gaussian. In contrast, the residual process for AR(1) model have been chosen Gaussian. Thus, the DAR(1) would naturally result in a better match than an AR(1) model with Gaussian residual if the data sequences were

intrinsically non-normal.. However, if similar residual sequences are used in both models, the difference between the two models is less subtle. The results have shown that a two-state Markov chain would not provide enough accuracy in the model and larger number of states was required. Furthermore, the model has proven to be effective when several sources are multiplexed, but is not effective to emulate a single video source.

3.2. GAR model

As pointed out in the previous section, Heyman *et al.* [9] have shown that the number of ATM cells per frame of VBR video conference follows a Gamma distribution. But, in their video source model, it was assumed that the residual is normally distributed. Thus, naturally, the number of cells per frame x_n generated from them would be normally distributed, and hence the model did not correctly model the occurrence of frames with large number of cells. One drawback of normal residual is that it takes negative values and produces the number of cells per frame x_n of negative cells. One way of overcoming this problem is to define $x_n = 0$ if x_n takes a negative value. Another way is to use a Gamma distribution residual process instead of a normal distribution in the AR process. Xu *et al*. [10, 11] analyzed a first order Gamma AR (GAR) process with Gamma distribution residual process. The GAR model is defined as:

$$\phi_e(s) = a^3 + 3a^3(1-a)\frac{\beta}{\beta+2} + 3a(1-a)^2\left(\frac{\beta}{\beta+s}\right)^2 + (1-a)^3\left(\frac{\beta}{\beta+s}\right)^3 \tag{13}$$

where $\phi_e(s)$ is the Laplace transform of the residual process, a is the AR coefficient, and β is the scale parameter. The difference equation generating the series $\{x_n\}$ from this model is written as:

$$x_n = a \cdot x_{n-1} + e_n = \begin{cases} a \cdot x_{n-1} \\ a \cdot x_{n-1} + E_n \\ a \cdot x_{n-1} + G_n(2) \\ a \cdot x_{n-1} + G_n(3) \end{cases} \tag{14}$$

where $\{E_n\}$, $\{G_n(2)\}$, $\{G_n(3)\}$ are an *i.i.d* sequence of exponential (β), Gamma (β,2), and Gamma (β,3) random variables, respectively. Thus, the residual process of the GAR model can be considered as a convex mixture of a degenerate random variable with mass zero, an exponential (β), a Gamma (β,2) , and a Gamma (β,3) distribution. The model generated sequences with nearly similar statistic characteristics (mean, variance,

distribution, and autocorrelation functions) with real video trace. The results showed that the GAR model outperformed the DAR model and generated sufficient frames with a large number of cells. One problem of the GAR model is that the residual process in (14) is too complicated to generate if the order of AR process is increased, and thus performed well only for the special case of first order AR process. Similar to DAR model, the GAR model can be used to create video traces for simulation, but can not be used for theoretical analysis models.

3.3. GBAR model

An important issue that has been raised in the literature in source modeling is that there may be a need for more than one model of VBR videoconferencing data. The GBAR model (Gamma Beta AR process) is one such model that is different from the previous models in that it is accurate, easy to simulate, and suitable for modeling a single video source [12]. The GBAR model was first introduced by McKenzie [13] and later by Heyman who validated its use as a source model for VBR videoconferencing. The basic idea of GBAR model is to assume that AR coefficients A_n are Beta distributed and the residuals B_n are Gamma distributed. Thus, for the first order AR process X_n:

$$X_n = A_n X_{n-1} + B_n \tag{15}$$

the sum of independent Gamma random variables is also a Gamma random variable, and the product of independent Beta and Gamma random variables is a Gamma random variable. Simulating the GBAR process only requires the ability to simulate *i.i.d* Gamma and Beta random variables. The parameters of the Gamma and Beta distribution can be estimated from the real video traces by matching the mean and variance. The GBAR model was simulated using three different videoconference sequences and the mean queue length and cell loss rates were found to be close to the real video data. The GBAR model (similar to GAR model) does rely on generating Gamma residuals, however, as argued in the literature that the closure property does not apply to Gamma distribution, and thus the linear operation performed by an AR model fails to produce a Gamma traffic. Although the GBAR model is shown to be more accurate than the DAR model, it is not suitable for studying admission control algorithms particularly in ATM networks.

3.4. C-DAR model

The models discussed so far were based on simulation. Xu *et al.* [14] proposed an analytical approach to model video conference traffic based on the DAR model, called a continuous time-discrete state AR (C-DAR). Basically, the C-DAR model is a continuous Markov model and may be looked as a Markov modulated rate process which is mathematically tractable. Its transition rate matrix is:

$$\boldsymbol{Q} = f(\boldsymbol{P} - \boldsymbol{I}) \tag{16}$$

Here, $\boldsymbol{P}$ is the transition matrix of the DAR model as given by (3.12) and f is given by:

$$f = \frac{(ln\ \boldsymbol{P})}{(\rho - 1)\boldsymbol{T}} \tag{17}$$

where $\boldsymbol{T}$ is the state-transition time slot of corresponding discrete-time Markov chain. The C-DAR model has the same steady-state probability distribution and exponential autocorrelation function as the DAR(1) model. The model has M states and a vector $\mathbf{V}=(V_1, V_2, \ldots\ldots, V_i)$, where V_i is the cell rate in state *i*. Then the traffic can be expressed as (**Q**,**V**). The steady state cell loss rate (CLR) in closed form is thus given by:

$$\mathrm{CLR} = \sum_i \frac{(V_i - C)\,P_i}{\overline{V}} \tag{18}$$

where C is the queue output rate. The analysis of the C-DAR model was compared with the simulation and the model works fairly well. However, since the C-DAR model is similar to the DAR model, the C-DAR model suffers from the same drawback of the DAR model.

Another interesting analytical model is described in [31]. In this model, an exact analysis of a discrete-time queuing system driven by a discrete AR model of order 1 (DAR(1)) characterized by an arbitrary marginal batch size distribution and a correlation coefficient is provided. Closed-form expressions for the probability generating function and mean queue length are derived. The system performance of this model is quite sensitive to the correlation of the arrival process. The model is compared with the traditional Markovian process and is shown that the arrival processes of DAR(1) type exhibit larger queue length as compared with the traditional Markovian processes when the marginal densities and correlation coefficients are matched.

3.5. General AR model

A general AR model constitutes a versatile class of models that is capable of generating Gamma-distributed traffic with arbitrary correlation, model order, and shape parameter while retaining the computational efficiency of Gaussian simple AR models. The trick part of this model is to utilize a $\chi^2(1)$ sequence. Basically, the $\chi^2(1)$ has two features: 1) a $\chi^2(1)$ can be easily obtained from a Gaussian time series, which in turn can be efficiently generated using an autoregressive process; 2) a linear combination of independent $\chi^2(1)$ variables can be accurately approximated by a Gamma variable. Based on these two features, Zhang [15] has recently proposed the general AR model as a viable modeling approach for video conference traffic in ATM networks. The model consists of two steps: first the model decomposes the given Gamma process into a weighted sum of a number of $\chi^2(1)$ sequences; second, each

$\chi^2(1)$ sequence is obtained by squaring a Gaussian process, which is efficiently generated by using an AR model from the given covariance matrix. Let $\boldsymbol{y}$ be the generated Gamma VBR video traffic of N samples, which follows a joint Gamma distribution with covariance matrix $\boldsymbol{R}_y$, correlation matrix $\boldsymbol{C}_y$, shape parameter m and variance σ_y^2. Then, $\boldsymbol{y}$ can be defined as:

$$\boldsymbol{y} = \sum_{k=1}^{K} \alpha_k \, z_k \tag{19}$$

The set of $\{\alpha_k\}$ values can be determined by solving the following two simultaneous equations:

$$\sum_{k=1}^{K} \alpha_k = \sqrt{m\sigma_y^2} \tag{20}$$

$$\sum_{k=1}^{K} \alpha_k^2 = \frac{\sigma_y^2}{2} \tag{21}$$

and K can be determined from the following relation:

$$\mathrm{K} = \mathrm{floor}(\frac{m}{0.5}) + 1 \tag{22}$$

The sequence $\{ z_k \}$ are vectors that synthesizes the Gamma sequence and are mutually independent with identical distribution. The sequence z_k can be easily generated from a Gaussian sequence $\{ x_k \}$ with a zero mean and covariance matrix $\boldsymbol{R}_x$ by squaring $\{ x_k \}$. That is:

$$z_k = [x_k^2(1), x_k^2(2), \ldots\ldots\ldots\ldots x_k^2(\mathrm{N})] \tag{23}$$

The sequence { x_k } can in turn be obtained from the Gaussian AR models as given by Eq. (1) and using the relation $\boldsymbol{R}_x = \sqrt{\boldsymbol{C}_y}$. Flowchart of the general AR approach is given in Fig. 1.

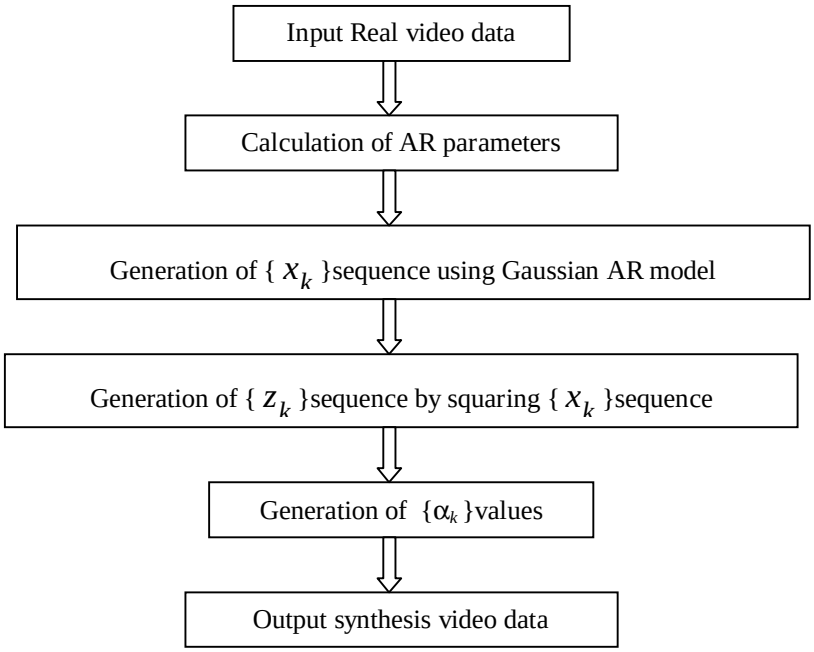

Fig. 1. The general AR algorithm.

The general AR model was generated using the same parameters values and AR(2) model as that described in Heyman [9]. The model produced frames sizes with more realistic and accurate values better than AR(2) and DAR(1) models. However, comparison with the real video traces in terms of mean, distribution, and autocorrelation function was not given and the model was not used in the simulation of a multiplexer. The difficult part of this model is to use trial and error to solve the two equations in (20) and (21) and then determine the appropriate real values of {α_k}. Further insight to this research point is worthy investigation.

4. Full-motion Video Models

Full motion video, unlike teleconference video, does not restrict its attention to scenes of people talking, but rather exhibits a wide range of video scenes (low, medium, and high) and includes background and foreground with frequent scene changes as in general TV program, news, and sports broadcast. This sort of video requires algorithms that produce higher quality output images than teleconferencing applications. Such algorithms rely on two sophisticated basic techniques: block-based motion compensation and discrete cosine transform based compression such as MPEG video. The use of motion compensation allows to reduce the temporal redundancy in the video sequence. In contrast to the teleconference video where motion compensation technique applies only causal coding (such as DPCM), full motion video applies both causal and non-causal coding. The use of DCT is to reduce spatial and perceptual redundancy. A full motion video sequence contains scenes, frames, and slices each corresponding to a different time scale. The fluctuations in the overall scene activities produce a series of variable bit rate sequences. Scenes containing a high degree of movement generate frames and slices at a high bit rate. Similarly, scenes containing a lower degree of movement generate frames and slices at a low bit rate. The highest bit rates arise during scene changes and last for few mille seconds. This produces video traffic with different statistical characteristics during different motion periods and higher bit rate relative to teleconference video traffic. Within each motion period, there is a strong correlation between the bit rates of successive frames. This motivated researchers to use an AR process as a model for full motion video. However, full motion video can not be represented by a single AR process and more elaborate models that capture the statistics associated with different time scale present in the video sequence and coding schemes are required. This section will discuss source video models that have been presented in the recent literature studies to model dynamic video sequences; namely motion classified AR, composite AR, scene based, PAR, three-layer AR, nested AR, GBAR GOP, and NAR. Advantages and disadvantages of each model and scheme are presented.

4.1. Motion classified AR model

Motion classified AR model was one of the early models proposed by [16] to model the output of a VBR codec by a first-order Gaussian AR process whose parameters are determined by the state of a finite Markov chain. The model works by classifying the observed data into scenes. The scene typically consists of tens or hundreds of frames that depict a particular real world scene. The scene is classified into three states: low, medium, and high activity scenes respectively, depending on the bit rate generation for the scenes according to thresholds. These thresholds are selected by visual inspection of the bit rate histogram of the actual video trace. Based on this scene classification, the scenes can be modeled by a three-state Markov chain where each state represents the degree of motion activity (low, medium, high). Let $\{S(n): n=1, 2, \ldots\}$ be a sequence of states. Then $S(n)=1$ denotes the Markov chain for the low activity scene, $S(n)=2$ for the medium activity scene, and $S(n)=3$ for the high

activity scene. The transition probability matrix of the Markov chain is found by measuring the average duration of each state and the number of transitions among different states. The bit generation during each state is modeled by independent AR(1) process. Let {X(n): n=1, 2, ….} be the number of bits in a frame, then the AR model can be expressed as:

$$X(n) = \begin{cases} a_i X(n-1) + e(n) & \text{if } S(n) = S(n-1) = i \\ b(n) & \text{if } S(n) \neq S(n-1);\ S(n) = i \end{cases} \quad (24)$$

where e(n) denotes a residual Gaussian random variable with specified mean and variance, a_i is the AR coefficient, and b(n) denotes another residual Gaussian random variable with specified mean and variance. The random variable b(n) is used to generate the bits after a state change. The parameters of the model can be obtained from the real video trace. The model requires a total of 9 parameters, three for each state. A schematic diagram of the model is shown in Fig. 2. The model has been tested using full motion color video sequence of 500 frames encoded by DCT and a motion compensation interframe using DPCM scheme using software implementation of motion adaptive coding algorithm. The mode has been shown to capture reasonably well the statistics of the bit rate. The difficult part of this model is to select the appropriate thresholds; a task which is not trivial. Similar study has also been reported in the literature, but using MPEG video traces [17]. In this study, the Markov chain was employed in which each state represents I, B, and P frames. The scene was classified based on the collection of GOPs. The scenes were identified by detecting I frames scene changes using a second difference method (will be described later in this section).

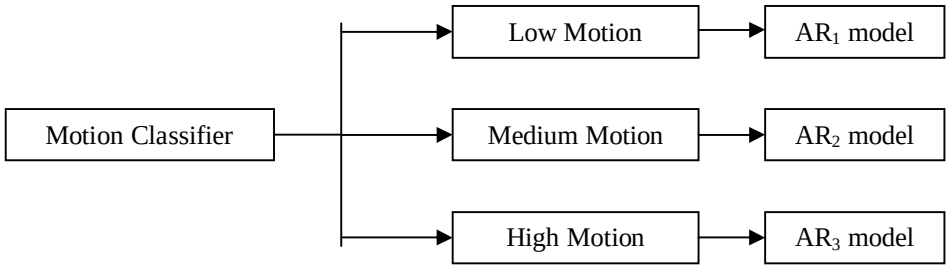

Fig. 2. schematic diagram of motion classified AR model.

4.2. Composite AR model

Ramaurthy *et al.* [18] have developed a model that involves the use of three stochastic processes. The first two processes use first-order Gaussian AR to capture the short term and long term autocorrelation of the bit rate. Short term correlations refer to the dependence of the bit rate of two successive frames, while long term correlations refer to the change in the bit rate characteristics from scene to scene. The third process uses Markov chain to incorporate the extra bits generated during scene changes. The sum of these three processes gives the following model:

$$R_n = X_n + Y_n + Z_n \tag{25}$$

where

$$X_n = a_1 X_{n-1} + A_n \tag{26}$$

$$Y_n = a_2 X_{n-1} + B_n \tag{27}$$

$$Z_n = K_n C_n \tag{28}$$

R_n is the generated bit rate video sequence (bits/frame) measured at the *n*th frame. X_n and Y_n are two AR processes used to achieve a better fit to the empirical autocorrelation at short lags and long lags, respectively. A_n and B_n are *i.i.d* normally distributed random variables with means μ_1 and μ_2 and variances σ_1^2 and σ_2^2. Z_n is a product of two terms K_n and C_n. K_n is a three-state Markov chain { K_n , n=1,2, …}whose states are 0, 1, 2 and its transition probability matrix *P* is given by:

$$P = \begin{bmatrix} 1-p & 0 & p \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{29}$$

C_n and *i.i.d* normally distributed random variables with mean $\alpha/2$ and variance $\beta/2$ that depend on K_n. The random process Z_n in Eq. (28) is introduced to capture sample path spikes due to video scene changes. Authors observed that scene changes last for two frames and the first frame after a scene change has significant more bits than other frames. Therefore, most of the time the Markov chain stays in state 0 and no extra bits are added. If this state is left, it will take two frames to get back to state 0. During this period the bit rate is increased. This model requires a total of 6 parameters ($\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \alpha, \beta$). The model was used to study the performance of a video multiplexer that multiplexes several full motion video sources. By suitable choice of model parameters, the model was shown to represent a variety of video application from low to high bit rate with large movement and scene changes. The

obtained data sets for this model was encoded by intraframe/interframe DPCM coding scheme.

4.3. Scene based AR model

The previous two models were studied using intraframe/interframe DPCM compression schemes. These coding schemes generate traffic of low to moderate bit rate and thus these models are not applicable to video movies. MPEG is a compression standard that has recently gained a considerable attention and widely used for video movies. Accurate MPEG source models are needed to support high speed networks such as ATM and Internet. Unlike other compression techniques, the MPEG video traffic exhibits periodic correlation structure and the bit rate distribution is quite complex as shown in Fig. 3. This is due to the underlying GOP (Group of Pictures) structure comprising an alternating pattern of the three frames I, B, and P with different average sizes and properties. These three types of frames are merged in a deterministic way to form the aggregate MPEG video sequence. Therefore, it is impossible to model MPEG video sources without considering its frame types. Most importantly, the traffic of an I frame subsequence that uses intra-coding without reference to other frames varies rapidly with scene. Since I frames are the largest, incorporating the scene activity in their sizes results in capturing most of the actual impact of scene activity.

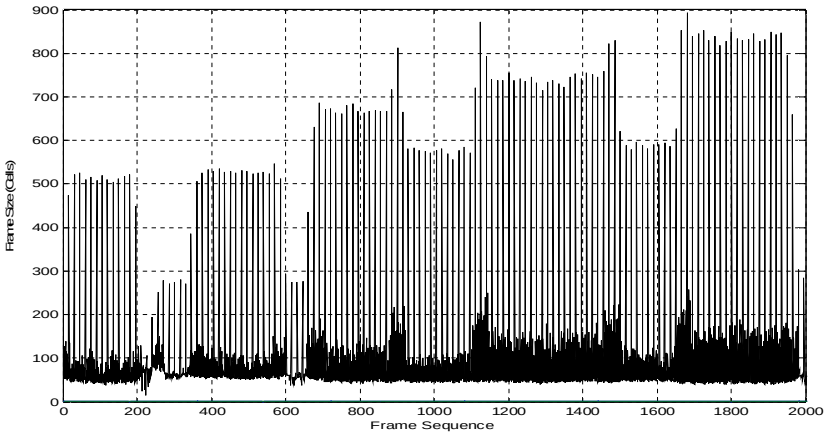

Fig. 3. Frame size trace of the Oz sequence.

Krunz *et al.* [19] proposed an MPEG traffic model containing I, B, P frames that uses a scene related component to capture bit rate variations at multiple time scales. First, the size of an I frame was modeled by the sum of two processes: the first is an *i.i.d* process used to generate the mean frame size of the scenes, and the second is an AR process used to generate the fluctuations within the scenes. The sum of these two random processes gives the following I frame model:

$$X_I(n) = M_I(n) + \delta_I(n) \tag{30}$$

where X_I(n) is the size of the *n*th I frame, M_I(n) is the mean of frame size of the scene to which the *n*th I frame belongs, and $\delta_I(n)$ represents the fluctuation of *n*th I-frame about the mean of the scene. Within the scene $M_I(n)$ will take the same value for every frame denoted by $\mu_I(j)$. Suppose the *kth* I-frame is the starting frame for the *jth* scene. Then the (*n*+*k*+1)*th* I-frame indicates the start of the next scene:

$$M_I(k) = M_I(k+1) = \ldots = M_I(k+N_j - 1) = \mu_I(j) \tag{31}$$

where $\mu_I(j)$ is lognormally distributed with mean μ_{μ_I} and variance σ_{μ_I} defined as:

$$f_{\mu_I}(u) = \frac{1}{u\sqrt{2\pi\sigma^2_{\mu_I}}} \exp\left[\frac{-(\ln u - \mu_{\mu_I})^2}{2\sigma^2_{\mu_I}}\right] \tag{32}$$

The scene change is identified as follows. Let X_I(n) be the number of cells in I-frame *n*th. At a scene change, the second difference (Diff2):

$$[X_I(n+1) - X_I(n)] - [X_I(n) - X_I(n-1)]$$

will be large in magnitude and negative in sign. In order to quantify the significant scene changes, the Diff2 is divided by the average of the past few frames as follows:

$$\frac{[X_I(n+1) - X_I(n)] - [X_I(n) - X_I(n-1)]}{\frac{1}{25}\sum_{j=n-25}^{n} X_I(j)} < -T \tag{33}$$

where *T* is the threshold. Using (31), the scene length is modeled as a geometric distribution given by:

$$g(x; p) = p(1-p)^{x-1} \tag{34}$$

where x = 1,2,3,… and *p* is a probability. The mean and variance of geometric distribution are μ = 1/p and $\sigma^2 = \frac{1-p}{p^2}$ respectively. The second process $\delta_I(n)$ is modeled as a second-order AR process:

$$\delta_I(n) = a_1\delta_I(n-1) + a_2\delta_I(n-2) + \varepsilon(n) \tag{35}$$

where $\varepsilon(n)$ is a sequence of *i.i.d* random variables with zero mean and variance σ_ε^2. The sizes of the P-frames $X_P(n)$ and B-frames $X_B(n)$ were modeled by two *i.i.d* random processes with lognormal distribution with means μ_{X_P}, μ_{X_B} and variances σ_{X_P}, σ_{X_B} respectively. The model was fitted very well to the real data. Figures 4 and 5 show a snapshot of the results. This model assumes that the sequence of the mean frame sizes $\{\mu_I(j)\}$ is modeled as a sequence of *i.i.d* random variables. We will see later a model proposed by Liu [25] which will replace the *i.i.d* random variables with an AR process. One drawback of the scene based AR model is that the scene related component is only used for modeling I frames and ignoring scene effects in P and B frames.

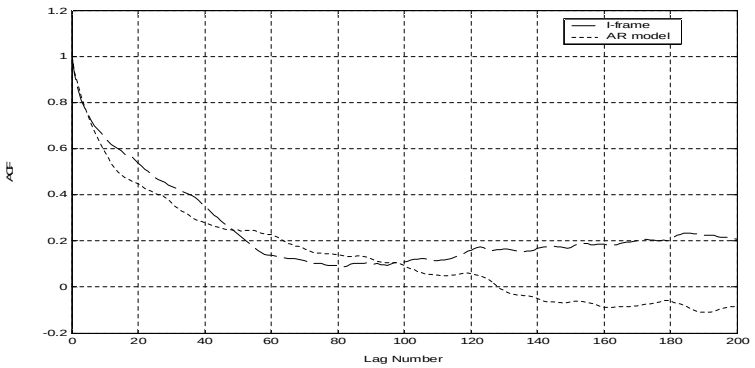

Fig. 4. Auto correlation function of real data and I-frame model for Star Wars sequence.

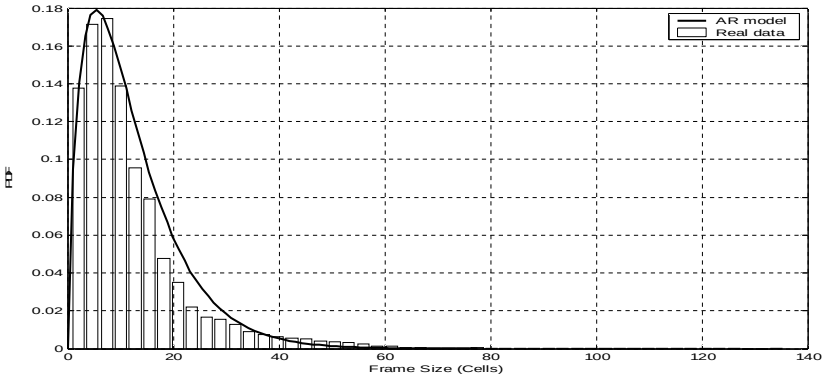

Fig. 5. Histogram of real data fitted to the synthesized scene based AR model for Star Wars sequence.

4.4. PAR model

Due to the bizarre and irregular histogram of video traffic, researchers have observed that a simple AR model can not fit the curve of histogram very well [7-9]. To compensate the AR model in fitting this bizarre histogram, a projected AR (PAR) model has been proposed at the cost of a slight deformation of the correlation properties. It was first introduced by Wu *et al.* [20] without considering the correlation of different types of MPEG frames, and then by Lien *et al.* [21] for the purpose of fitting the histogram of MPEG video traffic while preserving the autocorrelation property. Recently, an improved PAR(1) model was proposed matching the histogram and the short range dependency [22]. The basic idea of the PAR model is to project the data generated by the AR model to new ones in such a way that its distribution is closer to the real video. To project the data requires calculating, sampling, and storing the histograms of the real traces. To do so, the PAR modes uses CDF (cumulative distribution function) for the projection function. This is because the CDF statistically corresponds to the PDF (probability density function) and its monotonic increase function is very useful in the one-to-one projection. The empirical CDF of frame sizes can be obtained from the real video traces and the AR model. Let f denotes the frame size generated and takes the range between two consecutive points x_i and x_{i+1}. These two points can be obtained from the AR model by taking the normalized I, P, and B MPEG frame sizes ($X_I(n)$, $X_P(n)$, and $X_B(n)$) as the arrival for modeling, and is mapped to the CDF of the AR model to the cumulative probability p using the following linear equation:

$$p = y_i + (f - x_i)\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) \tag{36}$$

where y_i and y_{i+1} are the cumulative probabilities of x_i and x_{i+1} respectively. Let f` be the value on the CDF of the real video data with probability $p$. Then, by mapping $p$ onto the CDF of the real video data, $f$` gives:

$$f' = x_i' + (p - y_i')(\frac{x_{i+1}' - x_i'}{y_{i+1}' - y_i'}) \tag{37}$$

where y_i' and y_{i+1}' are the cumulative probabilities of x_i' and x_{i+1}' respectively. Values of x_i' and x_{i+1}' are obtained from the frame sizes $X_I(n)$, $X_P(n)$, and $X_B(n)$. The frame sizes generated by the PAR model can easily be obtained by dividing f` by the specified normalization factors. The PAR model produced sample data which was shown to match the histogram of the real vide data. However, the problem of this model is that the projection seriously affects the parsimony and transportability of the model.

4.5. Three-layer AR model

The autocorrelation function (ACF) of a typical MPEG video sequence presents periodical peaks with very slow decay, due to basically the fluctuations and dependency of the individual I, P, and B frames and GOP as shown in Fig. 6. This fluctuation of the ACF has convinced the researchers that such fluctuation can hardly be captured by a single AR process and led to the belief that a separate AR model for each frame type and GOP is required.

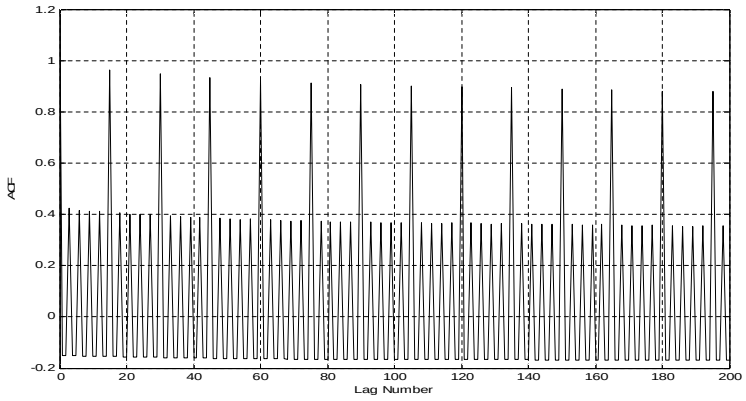

Fig. 6. ACF of a typical IBP MPEG sequence.

Doulamis *et al.* [23] analyzed the statistical properties of MPEG streams and developed three AR models of high order. The models concentrate on three layers: the frame layer, GOP layer, and intermediate layer. The frame layer model was introduced to achieve the correlation among I, P, and B frames; the GOP layer to provide an approximation of the aggregate MPEG sequence; and the intermediate layer to combine the properties of the frame and GOP layer for better and accurate approximation of MPEG traffic behavior. The summary of these three models are presented next.

4.5.1. Frame layer model

At frame layer, three high-order AR models were used for modeling I, P, and B frames (order of p = 25, 100, and 35 respectively). Let c∈ {I, P, B}, then the frame size is estimated using the equation:

$$X^c(n) = \sum_{k=1}^{p} a_i^c X^c(n-k) + e^c(n) \tag{38}$$

where $e^c(n)$ is the *i.i.d* Gaussian error. To ensure sufficient correrlated I, P, and B components, correlation of I, P, and B prediction errors was used to generate the model for the aggregate MPEG sequence. The method is to consider the error of B frames as reference error, due to the fact that B frames constitute the majority within a GOP. Then the error of I and B frames are related to that of B frames as follows:

$$e^I(n) = \begin{cases} e^{B_2}(n) & for\ odd\ n \\ e^I(n) & for\ even\ n \end{cases} \tag{39}$$

$$e^{P_i}(n) = e^{B_i}(n), \qquad i = 2,4,6 \tag{40}$$

The model was shown to generate MPEG traffic that match the real video trace and approximate the network resources. However, the model requires large number of parameters.

4.5.2. GOP layer model

Let $X^G(n_G)$ be the average frame size over a GOP period, where n_G denotes the GOP time series. Then, $X^G(n_G)$ can be generated from the AR model using the equation:

$$X^G(n) = \sum_{k=1}^{p} a_i^G X^G(n-k) + e^G(n) \tag{41}$$

Two methods were proposed to estimate the frame sizes I, P, and B by the knowledge of $X^G(n_G)$ within a GOP period (i.e. given $X^G(n_G)$, generate I, P, and B frames). The first method is to estimate the frame sizes to be equal to the respective value of $X^G(n_G)$ (i.e. the frame sizes are uniformly distributed within a GOP period). For example, if $X^1(1) = 10\,Kbits$ and the GOP is structured as: IBBPBBPBBPBB, then

$$X^I(1) = 10\,Kbits,$$
$$X^{P_1}(1) = X^{P_2}(1) = X^{P_3}(1) = 10\,Kbits$$
$$X^{B_1}(1) = X^{B_2}(1) = X^{B_3}(1) = X^{B_4}(1) = X^{B_5}(1) = X^{B_6}(1) = X^{B_7}(1) = X^{B_8}(1) = 10Kbits$$

This method assumes no knowledge about the properties of frame layer signal and can be used only for initial estimation of the network resources. The second method assumes that the mean values of I, P and B frames (denoted as μ^I, μ^P, μ^B respectively) are available . Then the estimated frames size is given by:

$$X^c(n_G) = \frac{\mu^c}{\mu^G} * X^G(n_G) \tag{42}$$

where c∈ {I, P, B} and μ^G is the mean value of signal $X^G(n_G)$. Like the first method, this method can be used for initial estimation of the network resources, but with more accuracy and better approximation.

4.5.3. Intermediate layer model

This model combines the significant parts of information of frame layer and GOP layer together to achieve a good approximation of MPEG traffic using a small number of parameters. The basic concept of the model is: 1) to simplify the GOP layer model so that only video activity is estimated; 2) to introduce simplified models for I, P, and frames based on the estimated video activity. To estimate video activity, each GOP is classified as belonging to one of three classes: high activity, medium activity, and low activity. The signal $X^G(n_G)$ is used for this classification. If $X^G(n_G)$ is greater than a threshold T_H, then GOPs are classified as high activity, those with $X^G(n_G)$ less than a threshold T_L, are classified as low activity and the rest are classified as medium activity. The thresholds are obtained using the mean and standard deviation of $X^G(n_G)$. The thresholds are chosen such that an autoregressive process models the temporal behavior of the autocorrelation function of the frames within a GOP. The video activity level of a GOP can be modeled as a Markov chain whose states

correspond to high, medium, and low activity states. Once GOP is classified as low, medium or high activity, then frames within a GOP are generated as follows. If GOP is low or medium class, set the frame sizes of I, P, and B to their mean values, since their exact values do not play a significant role to the traffic behavior. If GOP is high class, use three AR(1) models to generate I, P, and B frames respectively. This helps in reducing the complexity of the model. This model requires 21 parameters (6 parameters for Markov chain, 6 parameters for low and medium classes, and 9 parameters for high class).

4.6. GOP GBAR model

Frey *et al.* [24] developed a model for an MPEG video containing I, P, and B frames. The model called the group-of-pictures Gamma-beta autoregressive (GOP GBAR) model is an extension of Heyman's GBAR model [12] proposed for video teleconferencing. The GOP GBAR model explicitly accounts for MPEG GOP cyclicity and has convenient analytical properties with easily estimated parameters. Let X_B, X_P and X_I denote the sizes of MPEG B, P, and I frames, respectively. There frames can be modeled such that:

$$\begin{aligned} X_B &= Y_1 \\ X_P &= Y_1 + Y_2 \\ X_I &= Y_1 + Y_2 + Y_3 \end{aligned} \tag{43}$$

where Y_1, Y_2 and Y_3 are independent Gamma-distributed random variables; that is,

$$Y_i \sim Gam(\alpha_i, \lambda_i) \qquad i = 1, 2, 3 \tag{44}$$

with $\alpha_i > 0$ is the shape parameter and $\lambda_i > 0$ is the scale parameter. It is also possible to write:

$$\begin{aligned} X_B &= \lambda_1 Z_1 \\ X_P &= \lambda_1 Z_1 + \lambda_2 Z_2 \\ X_I &= \lambda_1 Z_1 + \lambda_2 Z_2 + \lambda_3 Z_3 \end{aligned} \tag{45}$$

where Z_1, Z_2 and Z_3 are independent standard Gamma random variables; that is,

$$Z_i \sim Gam(\alpha_i, 1) \qquad i = 1, 2, 3 \tag{46}$$

The GBAR model proposed by Heyman [12] for video teleconferencing is based on a recursion properly of Gamma and Beta random variables. Let *Be(p,q)* denote a Beta distributed random variable with parameters *p, q> 0*. It is shown that if:

$$\begin{aligned} Z_i(n-1) &\sim \text{Gam}(\alpha_i, 1) \\ B_i(n) &\sim \text{Be}(\alpha_i \rho_i, \alpha_i(1-\rho_i)) \\ W_i(n) &\sim \text{Gam}((1-\rho_i)\alpha_i, 1) \end{aligned} \tag{47}$$

where $0 \le \rho_i \le 1$, then:

$$Z_i(n) = B_i(n) Z_i(n-1) + W_i(n) \tag{48}$$

defines a stationary process $\{Z_i(n)\}$ with a marginal $Gam(\alpha_i, 1)$ distribution. Furthermore, the autocorrelation function of the process is given by:

$$r_i(k) = \rho_i^k \qquad k = 0, 1, 2, \ldots \tag{49}$$

Therefore, the size $X(n)$ of the *nth* frame in an MPEG-encoded video sequence starting with I-frame and using a *(N, M)* cyclic GOP can be modeled as:

$$X(n) = \begin{cases} \lambda_1 Z_1(n) + \lambda_2 Z_2(n) + \lambda_3 Z_3(n), & \text{if } n = 1 \bmod N, \\ \lambda_1 Z_1(n) + \lambda_2 Z_2(n), & \text{if } n \neq 1 \bmod N \text{ and } n = 1 \bmod M, \\ \lambda_1 Z_1(n), & \text{otherwise} \end{cases} \tag{50}$$

This model has six unknown parameters: $\{\lambda_i\}_{i=1}^3$, $\{\alpha_i\}_{i=1}^3$, *and* $\{\rho_i\}_i^3$. Under the assumption that Z_1, Z_2 and Z_3 are mutually independent, then:

$$\begin{aligned} \lambda_1 &= \frac{\sigma_B^2}{\mu_B}, \quad \lambda_2 = \frac{\sigma_P^2 - \sigma_B^2}{\mu_P - \mu_B}, \quad \lambda_3 = \frac{\sigma_I^2 - \sigma_P^2}{\mu_I - \mu_P}, \\ \alpha_1 &= \frac{\mu_B^2}{\sigma_B^2}, \quad \alpha_2 = \frac{(\mu_P - \mu_B)^2}{\sigma_P^2 - \sigma_B^2}, \quad \alpha_3 = \frac{(\mu_I - \mu_P)^2}{\sigma_I^2 - \sigma_P^2} \end{aligned} \tag{51}$$

where μ_B, μ_P, and μ_I are the sample means of the B, P, and I frames sequences, respectively, and σ_B^2, σ_P^2, and σ_I^2 are the corresponding sample variances. The remaining parameters ρ_1, ρ_2, and ρ_3 are estimated using the following formulas [3]:

$$\begin{aligned}\rho_1 &= \rho_B \\ \rho_2 &= \left(\frac{\sigma_P^2 \rho_P^M - \sigma_B^2 \rho_B^M}{\sigma_P^2 - \sigma_B^2} \right)^{1/M} \\ \rho_3 &= \left(\frac{\sigma_I^2 \rho_I^N - (\sigma_P^2 - \sigma_B^2)\rho_2^N - \sigma_B^2 \rho_B^N}{\sigma_I^2 - \sigma_P^2} \right)^{1/N}\end{aligned} \tag{52}$$

where ρ_B, ρ_P^M, and ρ_I^N are lag *1*, lag *M*, and lag *N* correlations of the B, P, and I frames, respectively. The model was fitted to six video frame sequences, and while close to those for the original video, do show some differences.

4.7. Nested AR model

Liu *et al*. [25] proposed a nested AR model which is a modified version of the scene based AR model [19]. Nested AR model takes scene changes into account and uses the hybrid Gamma/Pareto distribution for all three types of frames in MPEG-encoded video sequences. Specifically, the P and B frame types are modeled by two processes of *i.i.d* random variables with estimated parameters. However, the scene changes are incorporated in modeling the I frame sequence using two second-order AR processes nested with each other. One AR process is used to generate the main frame size of the scenes to model the long-range dependence, and another AR process is used to generate the fluctuations within the scene to model the short-range dependence. The parameters of the AR processes are estimated from measurement of empirical video sequences. Similar to the scene based AR model, let $x_I(n)$ be the size (i.e. the number of bits or cells) of the *nth* I frame in an MPEG video sequence. $x_I(n)$ will be modeled as the sum of two independent random variables:

$$x_I(n) = d(n) + f(n) \tag{53}$$

where *d(n)* is the mean frame size of the scene to which the *nth* I frame belongs and *f(n)* represents the fluctuation of the *nth* I frame about the mean of the scene. For the *jth* scene with length N_j that starts at the *kth* I frame, *d(n)* will take the same value for every frame within the scene, which will be denoted by another random variable $\tilde{x}_I(j)$, i.e.

$$d(k)=\frac{1}{N_j}\sum_{n=k}^{N_j+k-1}x_I(n)=d(k+1)=\ldots=d(k+N_j-1)=\tilde{x}_I(j) \tag{54}$$

The second random variable *f(n)* in (53) is used to fit the sequence obtained from the original data by subtracting the mean of the scene [i.e. *d(n)*] from each frame within the scene. This process actually eliminates the scenes. As a result, *f(n)* has zero mean and it models a "sceneless" sequence with a variance σ_f^2 very close, or equal to, the variance of the video traffic within scenes. The value of N_j can be determined using an approach similar to that given in section 4.3. To model the short dependence, a second order AR process is used for the sceneless sequence; that is,

$$f(n)=a_1 f(n-1)+a_2 f(n-2)+\varepsilon(n) \tag{55}$$

where {*ε(n)*} is a sequence of *i.i.d.* random variables. The long-range dependence is modeled using another second-order AR process for the mean sequence *d(n)*, or equivalently $\tilde{x}_I(j)$

$$\tilde{x}_I(j)=b_1\tilde{x}_I(j-1)+b_2\tilde{x}_I(j-2)+\theta(j) \tag{56}$$

where *{θ(j) }* is a sequence of *i.i.d.* random variables. Once the parameters of the two AR models are determined (using linear prediction method), the synthetic sequences {*f(n)*} and { $\tilde{x}_I(j)$ } are generated according to (55) and (56), respectively. The next step in this modeling approach is to obtain {*d(n)*} by combining { $\tilde{x}_I(j)$ } with the scene-length sequence. Scene-length distribution is well modeled by geometric distributions as given by (34). Similar to the scene based AR model, the P and B frame-size sequences { $x_P(n)$ } and { $x_B(n)$ } are generated by two processes of *i.i.d* random variables. Once generated, the I, P and B frame-size sequences are transformed to the hybrid Gamma/Pareto distribution. To transform a sequence { *s(n): n=0, 1, 2, …* } with Gaussian distribution F_G to a new sequence { *g(n): n=0, 1, 2, …* } with hybrid Gamma/Pareto distribution $F_{\Gamma/P}$, the following relationship is used [26]:

$$g(n)=F_{\Gamma/P}^{-1}\{F_G[s(n)]\} \tag{57}$$

The inverse of the cumulative probability function of hybrid Gamma/Pareto distribution $F_{\Gamma/P}^{-1}$ is computed as follows:

$$F_{\Gamma/P}^{-1}(y)=\begin{cases} F_P^{-1}(y)=\dfrac{a}{(1-y)^{1/\alpha}}, & \text{if } y>F_P(\hat{x})=1-\left(\dfrac{a}{\hat{x}}\right)^{\alpha} \\ F_\Gamma^{-1}(y) & \text{otherwise} \end{cases} \tag{58}$$

where F_P and F_Γ are the cumulative probability functions of Pareto and Gamma random variables, respectively. Note that F_Γ has no closed-form expression; hence, F_Γ^{-1} is obtained numerically. The function F_P, however, has the explicit form:

$$F_P(x)=\begin{cases} 1-\left(\dfrac{a}{x}\right)^{\alpha} & \text{if } x\geq a \\ 0 & \text{if } x<a \end{cases} \tag{59}$$

with parameters $a > 0$ and $\alpha > 0$ which are both determined by fitting. The parameter $\hat{x}$ in (56) represents the point at which the distribution of the empirical data deviates from the Gamma distribution. Specifically, the value of $\hat{x}$ can be estimated graphically by inspecting the tail of the empirical distribution, and determining where it starts to deviate from the tail of Gamma fit as shown in Fig. 7. The parameters of the theoretical Gamma distribution are obtained by matching the first and second moments of the empirical sequence to those of Gamma random variable. The part of the tail of empirical distribution which deviated from Gamma distribution is then fitted to the Pareto distribution. Mathematically,

$$F_{\Gamma/P}(x)=\begin{cases} F_\Gamma(x) & \text{if } x\leq\hat{x} \\ F_P(x) & \text{if } x>\hat{x} \end{cases} \tag{60}$$

Knowing the value of $\hat{x}$ and using the continuity condition $F_\Gamma(\hat{x})=F_P(\hat{x})$ along with least-square fitting of the Pareto tail, estimates of a and α can be obtained. Once the parameters a, α, and $\hat{x}$ are determined, the sequence $\{s(n)\}$ can be transformed into the new sequence $\{g(n)\}$ through the transformation defined in (57). Figure 8 shows a schematic diagram of the hybrid Gamma/Pareto distribution transformation. The nested AR model was compared to the scene based AR model using diverse contents of video sequence. Interestingly, the nested AR model gives rise to better autocorrelation at both small and large lags than the scene based AR model.

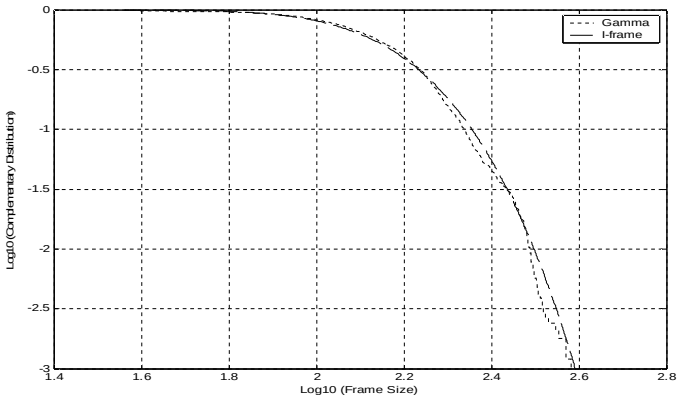

Fig. 7. Complementary frame size distribution for Movie2 sequence along with Gamma fit.

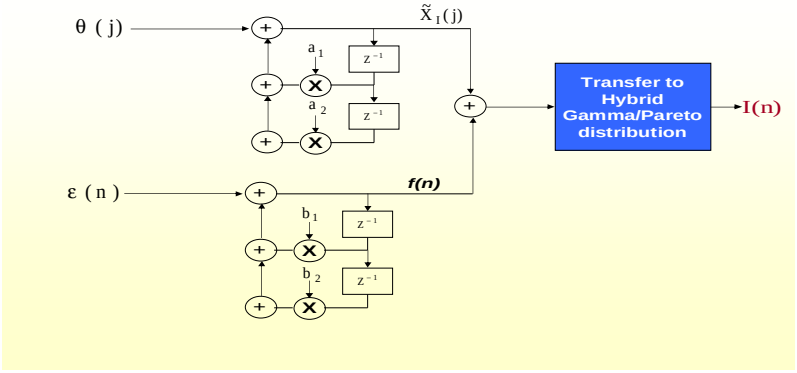

Fig. 8. A schematic diagram of the hybrid.

4.8. NAR model

Alheraish *et al.* [27] introduced a class of a non-linear AR (NAR) models that can be efficiently represented by quadratic filters which are widely used in modern digital signal processing. It was used to generate full motion MPEG video traffic at GOP layer. The NAR model is basically an extension of the general AR model proposed by Zhang [15] for videoconferecing. Similar to the general AR model, the NAR decomposes traffic data into a linear combination of a number of chi-square sequences, each of which is obtained by passing a Gaussian AR process through a simple non-linearity. However, the modeling approach of the NAR is based on second-order time-invariant filters. The

operator for such filters (denoted as H_2) is called second-order Volterra (or quadratic) operator. If $y(n)$ is the discrete time invariant Volterra filter whose input is $x(n)$, then

$$\begin{aligned} y(n) &= \mathrm{H}_2[\mathrm{x(n)}] \\ &= \sum_{l_1} \sum_{l_2} h_2(l_1, l_2)x(n-l_1)x(n-l_2) \end{aligned} \tag{61}$$

Let $z(n)$ be the GOP sizes to be modeled at the nth GOP. The sequence $z(n)$ can be represented as the sum of K mutually independent weighted sequences $\{y_k(n)\}$ that are generated by passing K independent Gaussian sequences $\{v_k(n)\}$ through K identical quadratic operators. That is:

$$\begin{aligned} z(n) &= \sum_{k=1}^{K} \alpha_k \mathrm{H}_2[v_k(n)] \\ &= \sum_{k=1}^{K} \alpha_k y_k(n) \end{aligned} \tag{62}$$

where $\{\alpha_k\}$ are constant weights and $\{v_k\}$ are zero-mean white Gaussian processes, each of which has variance σ^2. The objective is then to determine the kernels of optimum filter (or model) that best match the statistics of $z(n)$. Setting $h_2(i,j)=h(i)h(j)$ in (62) gives:

$$z(n) = \sum_{k=1}^{K} \alpha_k x_k^2(n) \tag{63}$$

where $x(n)$ is the output of a linear model with impulse response $h(i)$ and is given by:

$$x_k(n) = \sum_{li} h(i) v_k(n-i) \tag{64}$$

The weights $\{\alpha_k\}$ can be computed by matching the mean and variance of model's output to the mean and variance of real data. The sequence $x(n)$ can be modeled as an AR process of order p. The model was used to generate 3334 samples of GOPs and the sample realization, histogram and autocorrelation were compared to that of the real video data. In all cases, excellent matches were achieved except for the Star Wars movie which showed slight deviation. The queuing performance of the model showed very good match to the real video relative to the real video trace and a classical high order AR models, as demonstrated in Fig. 9. The model is only applied for GOP sequences and further study is thus required to account for cyclicity of the MPEG sequences.

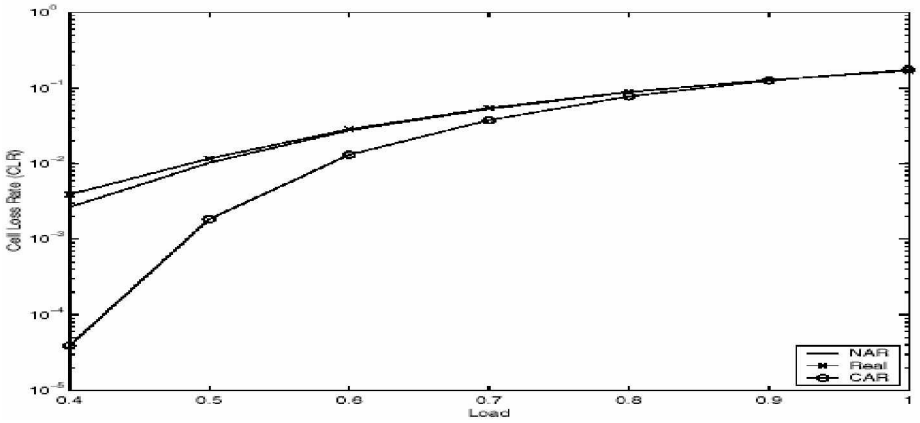

Fig. 9. Queuing performance of the NAR model relative to the real video trace and a classical high order AR models.

5. Summary, Recommendations and Open Issues

5.1. Summary

Video traffic modeling has become a key issue in the current literature due to the increasing importance of digital video services within multimedia and broadband networks. Video traffic models are needed to design networks that achieve acceptable picture quality at minimum cost, to control bit rate, to test the network performance (cell loss probability, end-to-end delays, jitter), and to evaluate call admission control/ bandwidth allocation. Of the several video traffic models reported in the literature, autoregressive models (AR) have received a considerable amount of effort. This is because the AR models have been shown to produce good results in capturing the bit and cell rate statistics. The coefficients for these models are simple to estimate from the empirical data using the autocorrelation coefficient. AR models, in general, appear to capture the autocorrelation behavior of compressed video sources which is an essential element for any model of compressed video sources. However, due to the nature of video traffic that exhibits dynamic and complex structure generated from different compression schemes, finding an appropriate AR model that can model different statistical characteristics is still an open problem.

Several good AR models have been studied extensively and many results have been reported. In this paper, we have examined, analyzed and compared state of the arts AR models that have appeared in the literature for modeling video traffic. Particularly, we have classified the AR models into two types: video conference models for modeling video conference and video phone like traffic with little motion (with no scene changes), and full motion video models for modeling broadcast video traffic with different motion levels (with sudden scene changes). Video conference

models are relatively simple to build as they require small set of parameters (mostly mean, variance, autocorrelation coefficient). On the other hand, the full motion video models are sophisticated and difficult to build. This is due to the complicated nature of this type of traffic. Table 1 and Table 2 provide a comprehensive summary of these two classes of models.

5.2. Recommendations

As a result of our investigations in this work, we found it is difficult to obtain an accurate (good) video model that represents dynamic nature of video traffic. In fact, there is no single video model is suitable for all video sequences and all purpose. As observed by Heyman [28], different sequences require different details regarding bit rate distribution. For instance, some sequences follow Weibul distribution, some follow Gamma, and others no simple model could be constructed. Also, we observed during our study that different sequences have different autocorrelation structures being at small lags or large lags. This is expected since different compression schemes (DPCM, JPEG, H.261, and MPEG) can lead to different bit rates for the same sequences. Even if an accurate model could be constructed, this would be a hugely complex, and perhaps more complicated than is necessary. In most cases, we seek to use a method that can satisfy our objectives and requirements for the purpose of network dimensioning and performance evaluation. In what follows, we suggest which of these video models should be considered for the analysis of a given communication network problem.

5.2.1. Video conference models

Out of the conference video models outlined in Table 1, GBAR and General AR models are the most appropriate approaches for queuing performance that hold for most feasible values of buffer sizes and service rates. Besides, these models mimic the statistical properties of the real traces. They are simple and easy to implement. Both models emulate single source. As pointed out by Lucantoni [29], single source models are useful for studying parameter negotiation, testing video rate control and predicting the quality of service degradation caused by congestion. General AR can even work for multiplexing several sources by modifying the parameters and weights of the model. DAR model may be a better choice if the model is intended for evaluating admission control or effective bandwidth. C-DAR gives the designer the simplicity to evaluate the performance of the network analytically and without having to do many simulation runs. The steady-state CLR and mean queue length are computed in closed form. If the model is just intended for an initial estimation of the network resources, then a simple AR model of first order would be sufficient to provide quick results or higher order AR if more accuracy is required.

Table 1. A summary of AR video conference models

Model	Type	Residual (error)	Video coding scheme	Queuing suitability	Limitations	Emulation	Ref.
Simple AR	AR(1) AR(2)	Gaussian	- DPCM - H.261	Simulation	Not accurate for modeling large number of cells	Single source	[7] [8] [9]
DAR	AR(1) + MC	Negative Binomial	DPCM	Simulation	Large number of Markov states is required/ Not suitable for single source	Several sources	[9]
GAR	AR(1)	Gamma	DPCM	Simulation	The residual process is too complicated to generate if high order AR is used	Single source	[11]
GBAR	AR(1)	Gamma	H.261	Simulation	The closure property does not apply to Gamma distribution/Not suitable for admission control	Single source	[12]
C-DAR	AR(1) + MC	Negative Binomial	-	Theoretical analysis	Same as DAR model	Several sources	[14]
General AR	AR with arbitrary order	Gaussian	DPCM	Simulation	The model works only if the bit rate distribution is Gamma./ Trial and errors are used to determine the weights of the model.	Single source	[15]

Table 2. A summary of AR Full motion video models

Model	Type	Residual (error)	Video coding scheme	Scene changes	Time scale	Limitations	Emulation	Ref.
Motion classified AR	AR(1) + MC	Gaussian	DPCM	No	Frame	Selecting appropriate thresholds is not easy task/ Complexity of the model increases if Markov states increases	Single source	[16]
Composite AR	AR(1)	Gaussian	DPCM	Yes	Frame	Complexity of the model increases if Markov states increases	Several sources	[18]
Scene based AR	AR(2)	Gaussian	MPEG	Yes	I, B, P frames	The scene related component is only used for modeling I frames and ignoring the scene effects in P and B frames	Several sources	[19]
PAR	AR(1)	Gaussian	MPEG	No	I, B, P frames	Projection is intended for matching histogram not correlation	Several sources	[21]
Three-layer AR	AR(1) With/ without MC	Gaussian	MPEG	NO	I, B, P frames/ GOP	Requires an extremely high order AR/large number of parameter	Several sources	[23]
GOP GBAR	AR (1)	Gamma	MPEG	NO	I, B, P frames/GOP	The closure property does not apply to Gamma distribution.	Single source	[24]
Nested AR	AR (2)	Gaussian	MPEG	Yes	I, B, P frames/GOP	Transformation using hybrid Gamma/Pareto does not model well long range dependence	Single and several sources	[25]
NAR	AR with arbitrary order	Gaussian	MPEG	No	GOP	The model works only if the bit rate distribution is Gamma./Trial and errors are used to determine the weights of the model/MPEG cyclicity is ignored	Single source and several sources	[27]

5.2.2. Full motion models

As stated in section 4, the MPEG syntax consists of 6 layers from top to bottom: sequence (scene), GOP, frame, slice, macroblock, and block. Ideally speaking, modeling MPEG full motion video could be done in any of these layers depending on the user's requirements and needs. During our course of study, we have found few AR models that deal with the bottom three layers slice, macroblock, and block (see for instance [5, 6]). Slice layer model can be used for studying transport packet payloads so that destination nodes can detect slice errors and provide correction action; macroblock and block models can be used for studying protocol data units in order to make a decision about the appropriate segmentation of the encoder output. However, modeling at these bottom layers in details would be complex and probably unnecessary task. On the other hand, modeling at the top three layers (scene, GOP, and frame) would be easier and simpler. Furthermore, since they reflect the multiple-time-scale variations of the bit rate, they are the most appropriate and strongly recommended for the analysis of a variety of communication networks requirements. GOP GBAR model is better than other models outlined in Table 2 when modeling frame layer since it is analytically tractable and has just small number of parameters. NAR model has convenient analytical properties and suitable for modeling GOP layer. One of the essential elements to be considered when evaluating full motion video models at these top three layers is the autocorrelation function. Most of the models presented in Table 2 are appropriate for modeling the autocorrelation function at small lags (or short range dependence SRD). Recent research results indicate that VBR video traffic exhibits large autocorrelation lags (long range dependence LRD) or persistence. LRD means the autocorrelation function is not summable, i.e. $\sum_{k=1}^{\infty} \rho_k = \infty$. Only nested AR model exhibits both SRD and LRD behavior and thus can be used to characterize video traffic and capture its correlations at multiple time scales. However, it is argued in the literature [12, 24], that LRD may not be necessary to incorporate in the dynamic model since the demand is determined by the purposes of the investigations. The target application in BISDN is also important in selecting the appropriate model and depends on the user's requirements. Some of these requirements are the following:

1) Initial estimation of network resources
2) Estimates of marginal distribution
3) Estimates of autocorrelation at both small and large lags
4) Prediction of network performance (packet/cell loss probability, delay and delay jitter)
5) Small or large buffers
6) Multiplexing homogeneous/heterogeneous sources

Based on these requirements, Table 3 summarizes the possible applications areas of each reviewed model.

Table 3. Recommended applications of AR full motion video models

Model	Target Application
Motion classified AR	1) Frame size prediction in a video coder 2) Low/medium video coders
Composite AR	1) Broadcast video with layered coding 2) Studying queuing length 3) Low/medium video coders
Scene based AR	1) Admission control algorithm 2) Congestion control 3) Multiplexer with a large buffer 4) Studying cell loss rate for homogeneous source
PAR	High accuracy of matching histogram of real video
Three-layers AR	1) Admission control algorithm 2) Multiplexer with a small buffer 3) Studying cell/frame/GOP loss rate 4) Prediction of video activity
GOP GBAR	1) Frame size prediction in a video coder 2) Studying frame loss rate 3) Studying parameter negotiation 4) Testing video rate control
Nested AR	1) Admission control algorithm 2) Congestion control 3) Studying short and long range dependency 4) Studying cell loss rate for homogeneous/heterogeneous source
NAR	1) Interaction of signal processing and source modeling 2) Studying scene activity level 3) Congestion control 4) Studying GOP loss rate 5) Model with adjustable parameters

5.3. Open issues

There remain many topics that deserve further investigations particularly for the recent models, such as:

1) Further study to general AR model by i) finding a suitable approach for evaluating the weights of $\chi^2(1)$ sequence, and ii) evaluating the queuing performance by multiplexing several sources and using a different video sequences.
2) If the general AR model and GBAR model are combined, does it produce better results?

3) Nested AR model is complex as it needs several processes to produce good results. In particular, the process of using transformation using hybrid Gamma/Pareto. Further insight to this point is worthy investigation.
4) Further work would involve evaluating Nested AR and NAR models using Wavelets instead of MPEG video coding scheme.
5) NAR model can be extended to model MPEG video at frame layer.
6) Since most of the bit rate distribution of the actual MPEG video traffic follows Gamma, the NAR model may be incorporated with the scene AR model to model B and P frames instead of the two *i.i.d* process. The mean frame sizes of the based AR model can also be modeled with the NAR approach.
7) Evaluating the presented AR models in Internet environment instead of ATM networks.
8) MPEG-2 is a two-layered coding. None of the AR models examined here has been modeled as a layered model. It would be interesting to explore this point of research.

Acknowledgment. The authors gratefully acknowlegde the Research Center at the College of Engineering, King Saud University, Riyadh, Saudi Arabia for its support of the project under the code number 423/15.

References

[1] Adas, A. "Traffic Models in Broadband Networks." *IEEE Communications Magazine*, 35 (July, 1977), 82-89.
[2] Krunz, M. "Statistical Multiplexing." *University of Arizona, Technical report No. TR-97-113*, 1994.
[3] Zquierdo, M. and Reeves, D. "A Survey of Statistical Source Models for Variable-bit Rate Compressed Video." *Multimedia Systems*, 7 (1999), 199-213.
[4] Casilari, E., Reyes, A., Diaz, A. and Sandoval, F. "Classification and Comparison of Modeling Strategies for VBR Video Traffic." *ITC-16*, 3b (June, 1999), Edinburgh, 215-221.
[5] Jabbari, B., Yegenolu, F., Kuo, Y. and Zhang, Y. "Statistical Characterization and Block Based Modeling of Motion Adaptive Coded Video." *IEEE Trans. Circuits Syst. Video Technology*, 3 (Feb., 1993), 199-207.
[6] Rye, B. and Elwalid, A. "The Importance of Long range Dependence of VBR Traffic in ATM Network Traffic Engineering: Myths and Realities." *Proceedings of ACM SIGCOMM'96 Conference*, (Aug., 1996), 87-92.
[7] Maglaris, M., Anastassious, D., Sen, P., Karlsson, G. and Roberts, J.D. "Performance Models of Statistical Multiplexing in Packet Video Communications." *IEEE Trans. On Comm*., 36, No. 7 (July, 1988), 834-843.
[8] Nomura, M., Fujii, T. and Ohta, N. "Basic Characteristics of Variable Bit Rate Video Coding in ATM Environment." *IEEE J. Selected Area Commun*., 7 (June, 1989), 752-760.
[9] Heyman, D., Tabatabai, A. and Lakshman, T.V. "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks." *IEEE Trans. Circuits Syst. Video Technology*, 2 (March, 1992), 49-59.
[10] Xu, S. and Hung, Z. "A Novel VBR Video Model on ATM Networks." In: *Proc. Int. Conf. Communications Technology*, May 5-7, 1996, Beijing, China, Vol. 2, pp. 1045-1048.
[11] Xu, S. and Hung, Z. "A Gamma Autoregressive Video Model on ATM Networks." *IEEE Trans. Circuits Syst. Video Technology*, 8 (April, 1998), 138-142.
[12] Heyman, D. "The GBAR Source Model for VBR Videoconferences." *IEEE/ACM Trans. Networking*, 5 (August, 1997), 554-560.

[13] McKenzie, E. "Autoregressive Moving Average Processes with Negative Binomial and Geometric Mariginal Distributions." *Adv. Appl. Prob*, 18 (1986), 679-705.
[14] Xu, S., Hung, Z. and Yao, Y. "An Analytically Tractable Model for Video Conference Traffic". *IEEE Trans. Circuits Syst. Video Technology*, 10 (Feb., 2000), 63-567.
[15] Zhang, Q. T. "A General AR-based Technique for the Generation of Arbitrary Gamma VBR Video Traffic in ATM Networks." *IEEE Trans. Circuits Syst. Video Technology*, 9 (Oct., 1999), 1130-1137.
[16] Yegenolu, F., Jabbari, B. and Zhang, Y. "Motion-classified Autoregressive Modeling of Variable Bit Rate Video." *IEEE Trans. Circuits Syst. Video Technology*, 3 (Feb., 1993), 42-53.
[17] Chiruvolu, G., Das, T., Sankar, R. and Ranganathan, N. "A Scene Based Generalized Markov Chain Model for VBR Video Traffic." *ICC 98*, (June, 1998), 554-558.
[18] Ramamurthy, G. and Sengupta, B. "An Analysis of a Variable Bit Rate Multiplexer Using Loss Priorities." *Computer Networks and ISDN*, 28 (Jan., 1996), 56-67.
[19] Krunz, M. and Tripathi, S. K. "On the Characterization of VBR MPEG Streams." *Perform. Eval. Rev.*, 25 (1977), 192-202.
[20] Wu, J., Chen, Y. and Jiang, K. "Two Models for Varaible Bit Rate MPEG Sources." *IEICE Trans. On Communications*, E78-B (May 1995), 737-745.
[21] Lien, J., Chen, Y. and Shiu, C. "Traffic Modeling and Bandwidth Allocation for MPEG Video Sources in ATM Networks." *Infocom*, 95 (1995), 2237-2241.
[22] Casilari, E., Reyes, A., Diaz, A. and Sandoval, F. "Scene Oriented for VBR Video." *Electronics Letters*, 34 (Jan., 1998), 166-168.
[23] Doulamis, N., Doulamis, A., Konstantoulakis, G. and Stassinopolos, G. "Efficient Modeling of VBR MPEG-1 Coded Video Sources." *IEEE Trans. Circuits Syst. Video Technology*, 10 (Feb., 2000), 93-112.
[24] Frey, M. and Nguyen-Quang, S. "A Gamma-based Framework for Modeling Variable Bit Rate MPEG Video Sources: The GOP GBAR Model." *IEEE/ACM Trans. Networking*, 8 (December, 2000), 710-719.
[25] Liu, D., Sara, E. and Sun, W. "Nested Auto-regressive Processes for MPEG Encoded Video Traffic Modeling." *IEEE Trans. Circuits Syst. Video Technology*, 11 (Feb., 2001), 169-183.
[26] Krunz, M. and Makowski, A. "Modeling Video Traffic Using M/G/∞ Input Processes: A Compromise between Markovian and LRD Models." *IEEE J. Select Areas Commun.*, 16 (June, 1998), 733-748.
[27] Alheraish, A. and Alsheibili, S. "A Gamma Source Model for Full Motion Video Using Non-linear Systems." *International Conference on Electronics, Circuits, and Systems*, UAE, 1 (Dec., 2003), 429-434.
[28] Heyman, D.A. and Lakshman, T.V. "Source Models for VBR Broadcast Video Traffic." *IEEE/ACM Trans. on Networking*, 4 (Feb., 1996), 40-48.
[29] Lucantoni, D., Neuts, M. and Reibman, A. "Methods for Performance Evaluation of VBR Video Traffic Models." *IEEE/ACM Trans. on Networking*, 2 (April, 1994), 176-180.
[30] Lazar, A., Pacfici, G. and Pandarakis, D. "Modeling Video Sources for Real-time Scheduling." *ACM/Springer Verlag, Multimedia Systems*, 1 (April, 1994), 21-32.
[31] Hwang, G. and Sohraby, K. "On the Exact Analysis of a Discrete-time Queuing System with Autoregressive Inputs." *Journal of Queuing Systems*, 2 (July, 2002), 44-57.

النمذجة الانحدارية
لحركة مرور الصور في الشبكات ذات النطاق الواسع

عبدالمحسن الهريش، صالح الشبيلي وطارق العمري

قسم الهندسة الكهربائية، كلية الهندسة، جامعة الملك سعود، ص.ب ٨٠٠،
الرياض ١١٤٢١، المملكة العربية السعودية

(قدِّم للنشر في ٢٠٠٣/١٠/٢٦م؛ وقبل للنشر في ٢٠٠٤/١٢/١٨م)

ملخص البحث. يشهد العالم في الآونة الأخيرة تطوراً هاماً نحو انتشار شبكات الخدمات الرقمية المتكاملة ذات النطاق الواسع مثل شبكة النقل غير المتزامن والتي لها المقدرة على دعم تقنيات الصور الرقمية وستشكل هذه التقنيات والتي منها على سبيل المثال فيديو عند الطلب ، فيديو المحادثة، فيديو الهاتف، فيديو البث، حركة مرورية هامة في هذه الشبكات وبالأخص الصور ذات البتات المتغيرة . أصبحت نمذجة الحركة المرورية منذ ما يزيد على عقد من الزمان مجالاً بحثياً نشيطاً، حيث اقترحت العديد من النماذج الجيدة لحركة مرور الصور ذات البتات المتغيرة. وسيركز هذا البحث بشكل رئيس على دراسة عدد من نماذج الإنحدار والتي تحتوي على خليط من تعقيدات الصور. وسنقسم نماذج الصور إلى نوعين: الصور ذات الحركة البطيئة والصور ذات الحركة السريعة. وفي كل نوع سنشير إلى المميزات الرئيسية، مع إعطاء وصف للنموذج وذكر لمزاياه وعيوبه. وسنقدم بعض النتائج للخصائص الإحصائية ومقدرة النموذج على التنبؤ بكفاءة الصفوف لسلسلة صور منفردة ومتعددة.