Long-Range Prediction for Real-Time MPEG Video Traffic: An H_∞ Filter Approach

Chih-Hu Wang, Bor-Sen Chen, *Fellow, IEEE*, Bore-Kuen Lee, *Member, IEEE*, Tsu-Tian Lee, *Fellow, IEEE*, Chien-Nan Jimmy Liu, and Chauchin Su, *Member, IEEE*

***Abstract*—A novel prediction scheme is proposed for real-time MPEG video to predict the burst and long-range dependent traffic. The trend and periodic characteristics of MPEG video traffic are fully captured by a proposed stochastic state-space dynamic model. Then a recursive H_∞ filtering algorithm is proposed to estimate traffic for long-range prediction. Simulation results based on real MPEG traffic data show that the proposed scheme has superior performance and lower complexity than some adaptive neural network methods, such as TDNN, NARX, and Elman neural networks.**

***Index Terms*—H_∞ filter, long-range dependence, MPEG video, state-space method.**

I. Introduction

In future broad-band communication networks, video applications are expected to be the major source of traffic data. The characteristics of video data for different applications have been the subject of many performance studies [1]. It has been shown recently [2] that most long video sequences exhibit long-range dependence. Furthermore, variable-bit-rate (VBR) video traffic can be nonstationary [3]. Due to the properties of long-term dependence and nonstationarity, it is still not easy to develop an effective prediction scheme for long-range prediction of VBR traffic.

Various prediction-based techniques have been proposed in recent years, for examples, recursive least square (RLS) [4], time delay neural network (TDNN) [4], [5], pipelined recurrent neural network (PRNN) [6], adaptive linear prediction [7], adaptive LMS algorithm for scene change detect [8], and recurrent neural network [9]. Comparison of various neural network architectures on prediction of MPEG traffic was made in [10]. Note that for the works in [5]–[10], off-line training of the neural networks is required in order to perform traffic prediction. Also, these techniques in [4]–[10] mostly decompose an MPEG trace into I, P, and B frame sequences and then predict these sequences separately. MPEG traffic representation using an $I-P-B$ composite statistical model was studied in [8], [11].

In this paper, by the periodic property of the MPEG encoded sequence, a composite MPEG traffic data is considered as a sequence consisting of a trend component, two periodic components, and a residual term, which can be fitted by a state-space model with these components being included in the state vector. Therefore, the traffic prediction problem becomes a state estimation problem. Since the statistics of the noises in the state-space model are usually unknown or uncertain, the trend and periodic components will be estimated by the robust H_∞ filtering method [12] for long-range prediction of the MPEG traffic.

The paper is organized as follows. In Section II, we shall show the characteristics of a MPEG video trace can be fully captured by a stochastic dynamic state-space model with traffic parameters in the state vector. With this model, traffic prediction is then transformed into a state estimation problem. In Section III, an H_∞ filter approach is presented to estimate the traffic parameters and predict the traffic. Simulation study is made in Section IV. Conclusions are given in Section V.

II. State-Space Signal Model for MPEG-Encoded Video Sequence and the Prediction Problem

An MPEG video stream is divided into the units called group of picture (GOP) which contain intraframe-coded frames (I-frames), interframe-coded frames (P-frames), and interpolative frames (B-frames). An example of the coding modes of a video sequence could be [1]

$$\underline{IBBPBBPBBPBB}\ \underline{IBBPBBP}\ldots\ldots \tag{1}$$

in which the period of the coding mode is $N = 12$. Hence, an MPEG video sequence $y(n)$ is dominated by a long-term periodic component (I frames) with a period $N = 12$ and a short-term periodic component (P frames) with a period $K = 3$. The remainder can be characterized by a local trend and a noise. Therefore, the t_d-step ahead video traffic consists of the following four components

$$y(n+t_d) = a(n+t_d)+c(n+t_d)+d(n+t_d)+w(n+t_d) \tag{2}$$

where $y(n+t_d)$ denotes the number of bits of the $(n+t_d)$-th frame, $a(n+t_d)$ the local linear trend component, $c(n+t_d)$ the

Manuscript received December 14, 2006; revised June 15, 2007 and October 23, 2007. First published September 16, 2008; current version published November 26, 2008. This work was supported in part by Chung Hua University under Grant CHU-95-ADV-002. This paper was recommended by Associate Editor W. Zhu.

C.-H. Wang is with the Department of Electrical Engineering, National Central University, Chung-Li, Taiwan, R.O.C., and also with the Departments of Communication Engineering and Electrical Engineering, respectively, Chung Hua University, Hsinchu, Taiwan 300, R.O.C. (e-mail: mike@chu.edu.tw).

B.-S. Chen is with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan 300, R.O.C. (e-mail: bschen@moti.ee.nthu.edu.tw).

B.-K. Lee is with the Departments of Communication Engineering and Electrical Engineering, Chung Hua University, Hsinchu, Taiwan 300, R.O.C. (e-mail: bklee@chu.edu.tw).

T.-T. Lee and C. Su are with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C. (e-mail: ttlee@cc.nctu.edu.tw; ccsu@mail.nctu.edu.tw).

C.-N. J. Liu is with the Department of Electrical Engineering, National Central University, Chung-Li, Taiwan, R.O.C. (e-mail: jimmy@ee.ncu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.2004926

long-term periodic component, $d(n+t_d)$ the short-term periodic component, and $w(n+t_d)$ the residual modeling error or noise. The prediction step t_d depends on application conditions such as the structure of the video, the transmission delay, and the delay in negotiating the rate with the network.

The local linear trend can be modeled as [13]

$$a(n+t_d) = a(n) + b(n)t_d + v(n) \tag{3}$$

where $a(n)$ denotes the trend at frame index n, $b(n)$ the trend slope, and $v(n)$ the driving noise. Since the trend slope $b(n)$ is random in practical video sequences, it can be modeled by the following random walk

$$b(n+1) = b(n) + u(n) \tag{4}$$

where $u(n)$ is an i.i.d. process with zero mean. Next, we consider the periodic components $c(n+t_d)$ and $d(n+t_d)$ in (2). From the long-term periodic component of an MPEG-encoded video sequence in (1), we have the following [13]

$$c(n+t_d) = c(n+t_d-N), \quad \sum_{t=n+1}^{n+N} c(t+t_d) = 0 \tag{5}$$

where N denotes the long-term period. Similarly, the short-term periodic component can be described by

$$d(n+t_d) = d(n+t_d-K), \quad \sum_{t=n+1}^{n+K} d(t+t_d) = 0 \tag{6}$$

where K denotes the short-term period.

For simplicity of analysis, the one-step ahead prediction model with $t_d = 1$ is discussed at first, i.e., we consider

$$y(n+1) = a(n+1) + c(n+1) + d(n+1) + w(n+1). \tag{7}$$

With (3)–(7), a state-space model for $y(n)$ is given as

$$\begin{aligned} X(n+1) &= FX(n) + BV(n) \\ y(n) &= GX(n) + w(n) \end{aligned} \tag{8}$$

where

$$\begin{aligned}
F &= \mathrm{diag}\{F_1, F_2, F_3\}, \; F_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\
F_2 &= \begin{bmatrix} -1_{1\times(N-2)} & -1 \\ I_{N-2} & 0_{(N-2)\times 1} \end{bmatrix} \\
F_3 &= \begin{bmatrix} -1_{1\times(K-2)} & -1 \\ I_{K-2} & 0_{(K-2)\times 1} \end{bmatrix} \\
B &= \begin{bmatrix} I_3 & \mathbf{0}_{3\times(N-2)} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times(K-2)} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times(N-2)} & 1 & \mathbf{0}_{1\times(K-2)} \end{bmatrix}^T \\
X(n) &= [a(n)\; b(n)\; c(n)\; c(n-1)\; \cdots\; c(n-N+2) \\
&\qquad d(n)\; d(n-1)\; \cdots\; d(n-K+2)]^T \\
V(n) &= [v(n)\; u(n)\; r(n)\; m(n)]^T \\
G &= \left[1\;0\;1\;\mathbf{0}_{1\times(N-2)}\;1\;\mathbf{0}_{1\times(K-2)}\right].
\end{aligned} \tag{9}$$

In the above equation, $I_{\overline{m}}$ is the $\overline{m}\times\overline{m}$ identity matrix, $\mathbf{1}_{1\times\overline{m}}$ is the $1\times\overline{m}$ vector with each entry being 1, and $\mathbf{0}_{\overline{m}\times\overline{p}}$ is the $\overline{m}\times\overline{p}$ matrix with all zero entries. Note that to allow $c(n)$ and $d(n)$ for random deviations from strict periodicity, noises $m(n)$ and $r(n)$ are added, respectively.

The next problem is to estimate the traffic parameters $X(n)$ from the video sequence $\{y(n)\}$ in (8). Once the estimate

$$\begin{aligned} \widehat{X}(n) = \Big[&\widehat{a}(n)\; \widehat{b}(n)\; \widehat{c}(n)\; \widehat{c}(n-1)\; \cdots\; \widehat{c}(n-N+2) \\ &\widehat{d}(n)\; \widehat{d}(n-1)\; \cdots\; \widehat{d}(n-K+2)\Big]^T \end{aligned}$$

is obtained, from (3), we can obtain the t_d-step ahead prediction of the local linear trend as

$$\hat{a}(n+t_d) = \hat{a}(n) + t_d\hat{b}(n) \tag{10}$$

and from the periodicity in (5) and (6), we get

$$\begin{aligned} \hat{c}(n+t_d) &= \hat{c}(n+t_d-N) \\ \hat{d}(n+t_d) &= \hat{d}(n+t_d-K) \end{aligned} \tag{11}$$

Therefore, from (2), we can get $\hat{y}(n+t_d)$ as follows:

$$\hat{y}(n+t_d) = \hat{a}(n) + t_d\hat{b}(n) + \hat{c}(n+t_d-N) + \hat{d}(n+t_d-K). \tag{12}$$

Now it remains to estimate $X(n)$ from the received sequence $y(n)$ based on the state-space model in (8).

III. Long-Range Traffic Prediction Based on H_∞ Filter

The covariance matrices of $V(n)$ and $w(n)$ are usually unknown beforehand. In this situation, a new approach based on the H_∞ filtering [12] is presented for state estimation without *a priori* knowledge of noise statistics. The measure of the H_∞ estimation performance is then given by

$$J = \frac{\sum_{n=0}^{M_0} \|e(n)\|^2}{\left\|X(0)-\hat{X}(0)\right\|^2 + \sum_{n=0}^{M_0}\left(\|V(n)\|^2 + \|w(n)\|^2\right)} \tag{13}$$

where $e(n) = X(n) - \widehat{X}(n)$, $X(0) - \hat{X}(0)$ represents the initial estimation error, and M_0 is the data size. The H_∞ filter will search the optimal estimation $\hat{X}(n)$ of $X(n)$ so that

$$\min_{\hat{X}(n)} \max_{\substack{V(n),w(n) \\ X(0)\in R^{N+K}}} J < r^2 \tag{14}$$

where $r > 0$ is a prescribed level of noise attenuation. The H_∞ filtering algorithm to solve our problem is given by [12]

$$R_e(n) = \begin{bmatrix} 1 & 0 \\ 0 & -r^2 \end{bmatrix} + \begin{bmatrix} G \\ L \end{bmatrix} P(n-1)[G^T \quad L^T] \tag{15}$$

$$H(n) = P(n-1)G^T\left(1 + GP(n-1)G^T\right)^{-1} \tag{16}$$

$$P(n) = FP(n-1)F^T + \beta^2 BB^T - FP(n-1)$$

$$\times [G^T \quad L^T] R_e^{-1}(n) \begin{bmatrix} G \\ L \end{bmatrix} P(n-1)F^T \quad (17)$$

$$\hat{X}(n) = F\hat{X}(n-1) + H(n)\left(y(n) - GF\hat{X}(n-1)\right) \quad (18)$$

where $H(n)$ is the gain of the H_∞ filter, $\hat{X}_0 = 0$, $L = [1\ 0\ \cdots\ 0]_{1\times(N+K)}$, and β^2 is a free design parameter. For the existence of the H_∞ filter, $P(n)$ needs to satisfy

$$P^{-1}(n) = P^{-1}(n-1) + G^T G - r^{-2} L^T L > 0. \quad (19)$$

After the video parameter estimate $\hat{X}(n)$ is calculated by the H_∞ filter, we can get the t_d-step ahead traffic prediction $\hat{y}(n+t_d)$ from (12).

IV. Simulation Study

In the simulation study, we shall demonstrate the superior prediction ability of the proposed H_∞ filtering algorithm by comparison with some adaptive prediction schemes based on different neural networks. The prediction performance of these algorithms are examined through comparison by simulation using various real MPEG-1 and MPEG-4 video traces, being available in the public domains[1] [14]. These chosen traces contain quite a diverse mixture of materials ranging from low complexity motion scenes to those with very high action. Before presenting simulation results, some prediction schemes based on dynamic neural networks are briefly described in the following.

A. Prediction by Using Dynamic Neural Networks

The dynamic neural networks [15], which are used for performance comparison, are briefly described in the following.

1) The focused time-delay neural network (FTDNN): We use an FTDNN with one input, 12 delay taps at the input, 24 neurons at the first layer, and one neuron at the output layer. The number of tuning parameters in the considered FTDNN is 360 including 312 weights and 48 bias constants.
2) The nonlinear autoregressive network with exogenous inputs (NARX): Here we use an NARX network with one input, 12 delay taps for the input, 12 delay taps for the output, 24 neurons at the first layer, and one neuron at the output layer. The number of tuning parameters in the considered NARX network is 648 including 600 weights and 48 bias constants.
3) The Elman network: We adopt an Elman network with 12 inputs, 24 neurons at the first layer, and one neuron at the output layer. The number of tuning parameters in the considered Elman network is 936 including 888 weights and 48 bias constants.

For the three networks, the nonlinear activation function at the first layer is the hyperbolic tangent sigmoid function and the activation function at the output layer is linear. In general, the nonlinear function between the input $y(n)$ and the output $g(n)$ of the considered neural networks can be defined as

$$g(n) = f\left(\phi_g(n-1), \phi_y(n-1), W, b\right)$$
$$\phi_g(n-1) = [g(n-1), g(n-2), \ldots, g(n-n_g)]$$
$$\phi_y(n-1) = [y(n-1), y(n-2), \ldots, y(n-n_y)]$$

where $n_y = 12$, W denotes the set of weights, and b is the set of bias constants. For the NARX network, we have $n_g = 12$; while for the other two networks, $n_g = 0$. Note that the settings of n_y and n_g are due to the fact that the period of the considered GOP structure is 12. The weights W and biases b are tuned in an adaptive manner. For a new arrival $y(n)$ of the MPEG traffic, W and b are tuned with one pass by using the scaled conjugate gradient backpropagation training method [15] to obtain $W(n)$ and $b(n)$ so that

$$[y(n) - f\left(\phi_g(n-1), \phi_y(n-1), W, b\right)]^2$$

is minimized. After tuning the weights and biases, the neural network can be used to recursively construct t_d-step ahead prediction of the MPEG\ traffic $y(n)$ by

$$\hat{y}(n+t_d) = f\left(\phi_g(n-1), \phi'_y(n+t_d-1), W(n), b(n)\right)$$

and $\phi'_y(n+t_d-1)$ is set as $[y(n), \ldots, y(n+1-n_y)]$, $[\hat{y}(n+t_d-1), \ldots, \hat{y}(n+1), y(n), \ldots, y(n+t_d-n_y)]$, and $[\hat{y}(n+t_d-1), \ldots\ldots, \hat{y}(n+t_d-n_y)]$ for the cases of $t_d = 1, 2 \le t_d \le n_y$, and $n_y < t_d$, respectively.

B. Comparisons of Prediction Performance

The parameters in the H_∞ filter are set as $N = 12$, $K = 3$, $\beta^2 = 3 \times 10^{-3}$, and $r = 10^5$. To evaluate the prediction performance, the performance index SNR^{-1}, defined by

$$\mathrm{SNR}^{-1} = \frac{\sum_{n=1}^{L_0} \left(y(n) - \hat{y}(n)\right)^2}{\sum_{n=1}^{L_0} y^2(n)},$$

is used where L_0 is the number of frames in a trace. We shall compare the prediction performance index SNR^{-1} under different prediction steps t_d ranged from 1 to 30 frames. Comparisons using the traces encoded by MPEG-1 and MPEG-4 are shown in Figs. 1 and 2, respectively.

The main computation burden of the H_∞ filter is due to the calculation of the covariance matrix $P(n)$. Let the number of entries in $P(n)$ be N_∞ where $N_\infty = (K+N)^2 = 225$ in our setting. Roughly speaking, the complexity of the H_∞ filter is $O(N_\infty^{3/2})$ at each adaptation step, while, in general, for the considered neural networks, the complexity is $O(N_n^2)$ with N_n being the number of parameters to be tuned in the network. Note that N_n is equal to 360, 648, and 936 for the FTDNN, the NARX neural network, and the Elman neural network, respectively. Actually, computation time for a neural network may further vary with parameter learning methods and possible line search methods. To make a fair comparison, the predictors based on the three neural networks are implemented by the standard function calls in the Matlab neural network toolbox [15] and the

[1][Online]. Available: http://www-info3.informatik.uni-wuerzburg.de/MPEG/

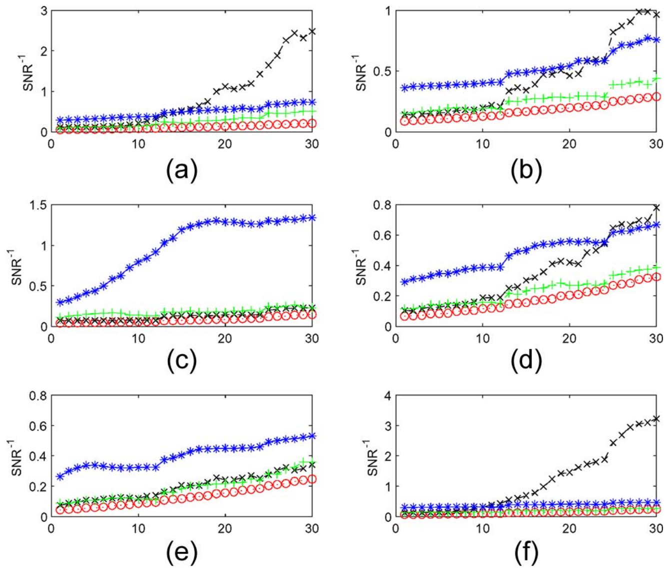

Fig. 1. SNR^{-1} prediction performance of the FTDNN ('x'), Elman ('*'), NARX ('+'), and the proposed H_∞ filter algorithm ('o') with respect to traces (a) *dino*, (b) *star*, (c) *mr.bean*, (d) *soccer*, (e) *race*, and (f) *atp* encoded by MPEG-1.[1]

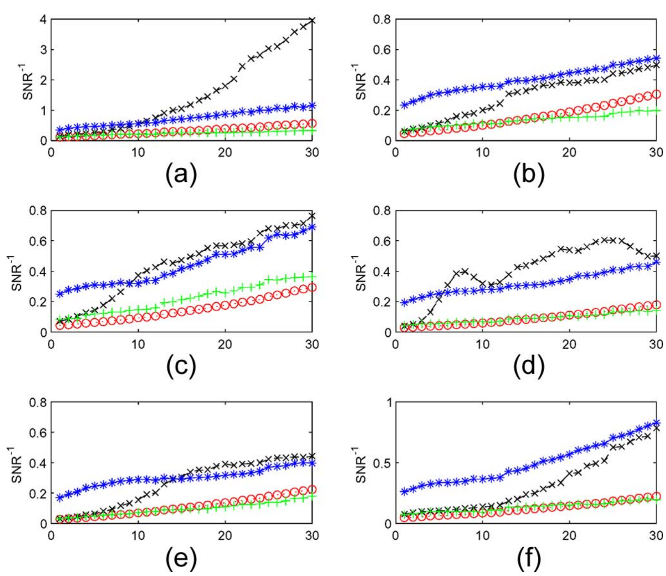

Fig. 2. SNR^{-1} prediction performance of the FTDNN ('x'), Elman ('*'), NARX ('+'), and the proposed H_∞ filter algorithm ('o') with respect to traces (a) *aladdin*, (b) *dusk*, (c) *contact*, (d) *Jurassic*, (e) *soccer*, and (f) *star4* encoded by MPEG-4 [14].

TABLE I
COMPARISON OF AVERAGE COMPUTATION TIME T_c IN SECONDS OF THE FOUR ALGORITHM AT EACH ADAPTIVE STEP WHERE R_T IS THE RATIO OF COMPUTATION TIME BASED ON THE H_∞ FILTER

	FTDNN	NARX	Elmam	H_∞
T_c	0.6197	0.6295	0.1334	9.222×10^{-5}
R_T	6719	6826	1446	1

complexity is evaluated by the computation time at each adaptation step. Complexity of these algorithm for making a prediction with $t_d = 30$ at each adaptation step are evaluated based on CPU execution time and are compared in Table I.

TABLE II
COMPUTATION TIME T_c OF A PREDICTION STEP UNDER VARIOUS PREDICTION HORIZONS

t_d	1	3	6	9	12
T_c (μs)	83.03	83.66	84.61	85.56	86.52

We also evaluate on-line computation time of the H_∞ filter by using the MATLAB software working on a personal computer with an Intel Core2 E6600 CPU. The average computation time of a prediction step under various prediction horizons is shown in Table II. For $t_d = 12$, the computation time at each time step is 86.52 μs which implies that to predict 100 MPEG video traces, it takes around 8.7 ms which is much shorter than a frame time interval (1/30 second).

From Figs. 1 and 2, it is obvious that the NARX neural network and the H_∞ filter have superior prediction performance to the other two algorithms. The SNR^{-1} value for the FTDNN scheme can be kept almost as low as that of the proposed H_∞ filter algorithm for $t_d \leq 10$ for most of the traces. On the contrary, as the prediction step t_d increases, the FTDNN scheme has a trend to diverge for a half of the considered traces. The proposed H_∞ filter and the NARX neural network are more robust in the sense that their SNR^{-1} performance indexes are more insensitive to the prediction step t_d than the other two algorithms.

For prediction of the MPEG traces encoded by the MPEG-1 technique as shown in Fig. 1, the H_∞ filter has the best performance. On the other hand, for those encoded by the MPEG-4 technique, the NARX prediction scheme is slightly better than the H_∞ filter for several traces. However, by comparing the computation time as shown in Table I, due to its simpler structure and lower computation burden, the H_∞ filter is better than the NARX scheme for real-time network traffic prediction with long-range delay to prevent traffic congestion in the multimedia transmission channel.

Remark 1: Normally, large prediction error comes from scene change. A recent algorithm to detect scene change can be referred to [16] and MPEG traffic prediction based on detection of scene change has been explored in [8]. However, this topic is more complicated and will be studied in the future.

Remark 2: Actually, a neural network with enough complexity can perform very good long-range prediction if a suitable off-line training procedure is carried out [5]–[10]. However, for bandwidth allocation at a network node, on-line adaptive prediction of the incoming MPEG traffic is required. Therefore, neural network should be in adaptive forms, in stead of being under the batch training mode. Moreover, to reduce computation time at each adaptation step, the learning epochs should not be large. These are the main reasons why some neural networks have unsatisfactory *adaptive* prediction performance.

V. CONCLUSION

In this study, based on the local linear trend, periodic components, and noise residue, a stochastic state-space model for MPEG video sequences is developed with traffic parameters included in its state vector for long-range prediction. The long-range prediction problem is thus changed to a state estimation problem. Then a robust H_∞ prediction algorithm is proposed without requiring statistical knowledge of the noises to achieve a precise long-range prediction of MPEG-encoded video traffic to compensate for long transmission delay. From simulation using real MPEG video traffic, the performance of the proposed scheme is verified by evaluating the prediction accuracy and the computation complexity.

REFERENCES

[1] S. C. Liew and D. C. Y. Tse, "A control-theoretic approach to adapting VBR compressed video for transport over a CBR communications channel," *IEEE Trans. Netw.*, vol. 6, no. 1, pp. 42–55, 1998.

[2] M. Grossglauser and J.-C. Bolot, "On the relevance of long-range dependence in network traffic," *IEEE/ACM Trans. Netw.*, vol. 7, no. 5, pp. 629–640, Oct. 1999.

[3] M. Grasse, M. R. Frater, and J. F. Arnold, "Testing VBR video traffic for stationarity," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 448–459, Apr. 2000.

[4] S. Chong, S. Li, and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 1, pp. 12–23, Jan. 1995.

[5] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "An adaptable neural-network model for recursive nonlinear traffic prediction and modeling of MPEG video sources," *IEEE Trans. Neural Netw.*, vol. 14, no. 1, pp. 150–166, Jan. 2003.

[6] P. Chang and J. Hu, "Optimal nonlinear adaptive prediction and modelling of MPEG video in ATM networks using piplined recurrent neural networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1087–1100, Aug. 1997.

[7] A. Adas, "Using adaptive linear prediction to support real-time VBR video under RCBR network service model," *IEEE Trans. Netw.*, vol. 6, no. 5, pp. 635–644, Oct. 1998.

[8] S.-J. Yoo, "Efficient traffic prediction scheme for real-time VBR MPEG video transmission over high-speed networks," *IEEE Trans. Broadcast.*, vol. 48, no. 11, pp. 10–18, Mar. 2002.

[9] A. Bhattacharya, A. G. Parlos, and A. F. Atiya, "Prediction of MPEG-coded video source traffic using recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2177–2190, Aug. 2003.

[10] A. Abdennour, "Evaluation of neural network architectures for MPEG-4 video traffic prediction," *IEEE Trans. Broadcast.*, vol. 52, no. 2, pp. 184–192, Jun. 2006.

[11] N. Ansari, H. Liu, Y. Q. Shi, and H. Zhao, "On modeling MPEG video traffics," *IEEE Trans. Broadcast.*, vol. 48, no. 4, pp. 337–347, Dec. 2002.

[12] B. Hassibi, A. H. Sayed, and T. Kailath, "Linear estimation in Krein spaces—Part II: Application," *IEEE Trans. Automat. Contr.*, vol. 41, no. 1, pp. 34–49, Jan. 1996.

[13] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*. New York: Springer, 1996.

[14] F. H. P. Fitzek and M. Reisslein, "MPEG-4 and H.263 video traces for network performance evaluation," *IEEE Network* vol. 15, no. 6, pp. 40–54, Nov.–Dec. 2001.

[15] H. Demuth, M. Beale, and M. Hagan, *Neural Network Toolbox 5: User's Guide*. Natick, MA: The MathWorks, 2007.

[16] H. Li, G. Liu, Z. Zhang, and Y. Li, "Adaptive scene-detection algorithm for VBR video stream," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 624–633, Aug. 2004.