MPEG VBR SLICE LAYER MODEL USING LINEAR PREDICTIVE CODING AND GENERALIZED PERIODIC MARKOV CHAINS

Michael R. Izquierdo and Douglas S. Reeves***

*Network Hardware Division
IBM Corporation
Research Triangle Park, NC 27709
mikeiz@raleigh.ibm.com

**Electrical and Computer Engineering
North Carolina State University
Raleigh, North Carolina 27695
reeves@eos.ncsu.edu

ABSTRACT

We present an MPEG slice layer model for VBR encoded video using Linear Predictive Coding (LPC) and Generalized Periodic Markov Chains. Each slice position within an MPEG frame is modeled using an LPC autoregressive function. The selection of the particular LPC function is governed by a Generalized Periodic Markov Chain; one chain is defined for each *I*, *P*, and *B* frame type. The model is sufficiently modular in that sequences which exclude *B* frames can eliminate the corresponding Markov Chain. We show that the model matches the pseudo-periodic autocorrelation function quite well. We present simulation results of an Asynchronous Transfer Mode (ATM) video transmitter using a FIFO queue and measure the average cell delay. Simulation results showed good agreement with results obtained using actual traces as sources.[1]

I. INTRODUCTION

The advent of video compression techniques has made it feasible to store and transmit video over computer networks. These techniques are embodied in international standards such as H.261, MPEG-1, and MPEG-2 [1,7,8]. This places us at the threshold of the next evolutionary turn of computer networks where the transport of compressed video will become commonplace. In order to accomplish this successfully, a greater understanding of how compressed video will impact the performance of future networks is needed. Researchers are using simulations to study important metrics such as packet loss, delay and statistical multiplexing gain. In order to do this effectively, adequate source models are required which capture the pertinent stochastic aspects of the video source.

The ATM Network has gained much attention as an effective means to transfer voice, video and data information over computer networks. ATM provides an excellent vehicle for video transport since it provides low latency with minimal delay jitter when compared to traditional packet networks [11]. As a consequence, there has been much research in the area of the transmission and multiplexing of compressed video data streams over ATM.

Compressed video differs greatly from classical packet data sources in that it is inherently quite bursty. This is due to both temporal and spatial content variations, bounded by a fixed picture display rate. Rate control techniques, such as CBR (Constant Bit Rate), were developed in order to reduce the burstiness of a video stream. CBR encoders control the output bit-rate by monitoring the level of an output buffer and adjusting video quality so that the buffer does not under or overflow. An alternate encoding method, called VBR (Variable Bit Rate), is seen as an alternative to CBR because it offers the possibility of increased statistical multiplexing gain (SMG) with minimal variations in video quality [12,20]. VBR does not control the output bit rate of the encoder and does not require an output buffer. Since the output bit is not controlled, the bit rate varies significantly with high peak-to-mean ratios (4:1, 6:1).

Many researchers have proposed VBR source models to use in simulations in order to quantify the amount of SMG. The majority of these models model the frame layer as opposed to the slice layer in that they generate the number of bits, bytes or cells contained within a frame[2] [2,3,5,9,17,21]. This was done for two reasons: (1) The Slice Layer exhibits an autocorrelation function which is pseudo-periodic which is difficult to model, and (2) Network buffering was assumed to be larger than a frame, thereby, minimizing the stochastic effects at time scales less than a frame period (slice). The second reason has led to the modeling of longer video

1. This work was supported by the Air Force Office of Scientific Research under Grant F49620-92-J-0441, and by an IBM Resident Study Fellowship.

2. Throughout the paper, we use the terms *picture* and *frame* synonymously.

sequences in order to capture the long range dependent (LRD) behavior of VBR video [2,3,6].

One reason for using slice layer model is for interactive real-time applications, with low delay requirements. In this case, the short range dependent (SRD) behavior of the autocorrelation function could be more important than the LRD behavior [14]. Also, a frame layer models cannot generate samples within a frame interval. Typically, if slices samples are needed the model must estimate the size of each slice by distributing the number of bits per frame either deterministically or randomly. This sacrifices accuracy affecting simulation results which could be particularly significant in a slice based packet delivery system.

We propose a slice layer model for MPEG which is based on Linear Predictive Coding (LPC) and Generalized Periodic Markov Chains (GPMC). Each slice is modeled using an LPC autoregressive function; one per frame type. We modeled four VBR video sequences called: *Bike*, *Flowg*, *Tennis* and *Undersiege* which contained *I*, *B*, and *P* frame types and 150 frames in length. The model is not specific to a frame type in that a model consisting of I/P frames is done by removing the GPMC for B frames.

We organized the paper as follows. Section II and III give a brief review of MPEG encoding and an overview of past work in slice modeling. Section IV provides a description of each video sequence and their statistical results. Section V presents a description of the video model and section VI presents and discusses the simulation results. Finally, section VII concludes the paper.

II. REVIEW OF MPEG ENCODING

The Motion Pictures Expert Group (MPEG) published the standard document 11172 parts 1, 2 and 3 which define the encoding, decoding and multiplexing of digital audio and video at bit-rates under 1.5 mbps [8]. Part 1 describes the multiplexing and synchronization of audio and video elementary streams. Part 2 defines the encoding and decoding of video while part 3 does the same for audio. An excellent introduction to MPEG is given in [15].

An MPEG video stream is organized as a hierarchy of layers called: Sequence, Group of Pictures (GOP), Picture, Slice, Macroblock and Block. The Sequence Layer consists of a sequence of pictures organized into groups called GOPs. An example of a GOP is shown in Figure 1. There are three types of pictures defined in MPEG: *I*, *B* and *P*. *I* pictures use the discrete cosine transform (DCT) and do not use motion compensation. Both *B* and *P* pictures use motion compensation with *B* pictures using forward, backward, and bidirectional predictive coding and *P* pictures using only forward predictive coding. *I* pictures typically contain more bits than both *B* and *P* pictures. *P* pictures generally containing more bits than *B* pictures. The GOP frame sequence pattern is determined by two encoding parameters, *N* and *M*. *N* defines the distance between *I* pictures, whereas, *M* defines the distance between *P* pictures or *I*, *P* pictures. Each picture is segmented into slices, where a picture can contain one or more slices. An example of a particular slice format, used by the videos studied in this paper, is shown in Figure 2. Here, a slice covers a complete macroblock row which gives use a total of 15 slices per frame.

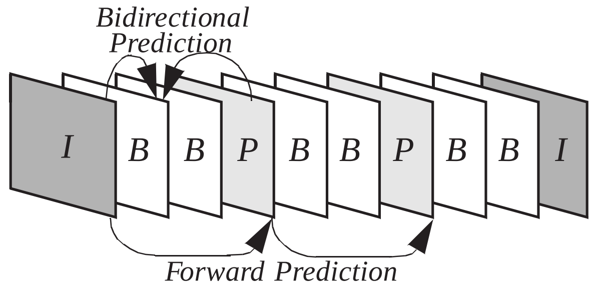

FIGURE 1. MPEG Group of Pictures.

Slices are independent decodable units in that, if corrupted, the decoder can decode the next slice without error. Each slice contains a sequence of macroblocks where a macroblock consists of four luminance blocks (Y) and two chrominance blocks (Cb and Cr)[1]. Each block is organized into a matrix of 8x8 pixel samples with a macroblock covering a 16 x 16 pixel area.

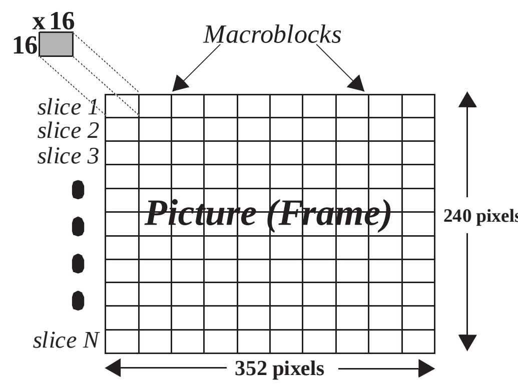

FIGURE 2. MPEG Frame structure.

III. SLICE LAYER MODELING

Researchers have proposed several slice layer models for VBR encoded video. Zdepski, et. al [22] used a 1st order synthesis lattice filter using a white noise input to model H.261 video conference sequences at the Group of Blocks (GOB) layer. The mean, variance, and autocorrelation at lag 1 where estimated from the actual sequences and used as model parameters. The model was used to simulate an 802.3 LAN network driven by H.261 video sources. Each data packet consisted of one or more GOBs (There are twelve GOBs in each frame). The model generated samples for the number of bits per frame and estimated the size of each GOB by dividing these bits equally for each GOB.

1. This is the 4:2:0 block pattern.

The previous model was improved by using a TES process in place of the lattice filter [18]. GOB estimation was accomplished by observing that the pseudo-periodicity of the GOB data was deterministic and could be removed by determining its fundamental frequency components. This was accomplished by using a periodogram. The periodicity was removed by subtracting it from the original sequence and TES was applied to the residual process. Once TES parameters were determined, the periodic process was added back in. The final model, therefore, consisted of a TES process and a deterministic periodic process.

Lazar, et. al. [14] used a Generalized TES process to model the slice layer of the Lucas films movie "*Star Wars*" which used an encoding similar to MPEG. The sequence was encoded using only DCT and consisted of blocks containing luminance values. They modeled the slice layer directly by adding a simple modulating function to the innovation function. Their model could be used for both the frame and slice layers. The slice model did produce an autocorrelation function which was pseudo-periodic.

TES has the advantage that it can match the distribution of the empirical data exactly and calculate the autocorrelation function quickly using the *TEStool* [4]. This approach does require the heuristic determination of parameters which will generate the appropriate autocorrelation function which can be a tedious and time consuming process. This has been alleviated somewhat by using an algorithm, based on minimum weighted distance, which automates the process of finding the appropriate TES parameters [10]. A simpler method which allows for the direct calculation of model parameters from statistical measurements would be more desirable.

Landry and Stavrakakis have shown that it is possible to generate a sequence in which the autocorrelation function is pseudo-periodic by using a Generalized Periodic Markov Chain [13]. We build upon this result to create a model whose parameters can be extracted directly from statistical measurements of the actual video data. We also desire a model which is modular in that it can be used with any combination of MPEG frame types.

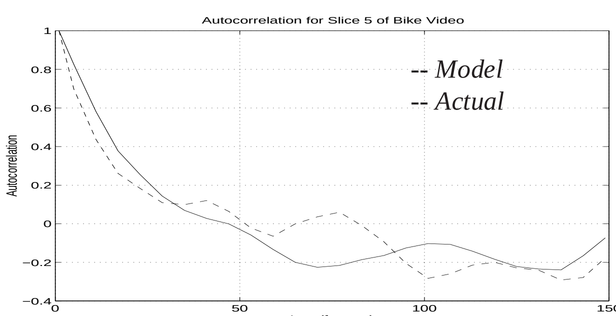

FIGURE 3. Autocorrelation Function for slice position 5 of Bike video.

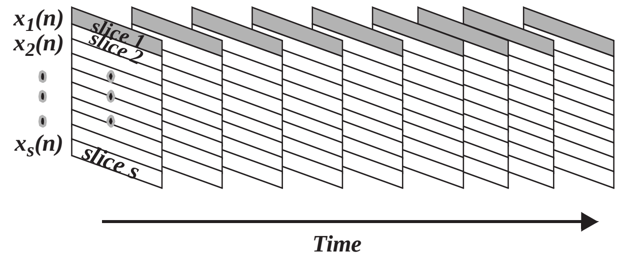

FIGURE 4. I-frame model by slice position.

IV. SLICE MODEL DESCRIPTION

The slice model is based on the observation, shown in Figure 3, that if we characterize the slice data based on its location within a frame and frame type the autocorrelation function is not pseudo-periodic and, in fact, exhibits significant correlation at lag 1 (Note, that lag 1 for an *I* frame slice indicates a correlation over *N* frames while for *P* frames, it is over *M* frames except where GOP boundaries occur). For this reason, we decided to use a model based on Linear Predictive Coding (LPC) where each slice position within each frame type is modeled using a first order autoregressive process (AR(1)).

Modeling by slice position has the advantage that it removes the effects of spatial content from the random process, leaving only temporal effects. This is illustrated in Figure 4 which shows separate LPC functions for each slice position. We used a different set of autoregressive processes for each frame type.

The LPC function consists of an all-pole filter fed by a white noise generator modulated by a constant gain factor. We used the first order LPC function given by

$$x(n) = ax(n-1) + Gv(n), \tag{1}$$

where *a* is the first order coefficient, *G* is the filter gain, and $v(n)$ is white noise. In order to determine *a* and *G*, we used a recursive method given in [19] where

$$a = \frac{\phi(1)}{\phi(0)}, \qquad G = \phi(0)(1-a^2). \tag{2}$$

The function, $\phi(n)$, is the biased estimate of the autocorrelation function for the sequence.

$$\phi(n) = \frac{1}{N}\sum_{i=1}^{N-n} x_i x_{i+n} \tag{3}$$

We define three sets of sequences: $x_i(n)$, $y_i(n)$, and $z_i(n)$ which consisted of the slice samples for each *I*, *P* and *B* frame, where *i* is the slice position within a frame and *n* is the frame number. We define the set of sequences to be:

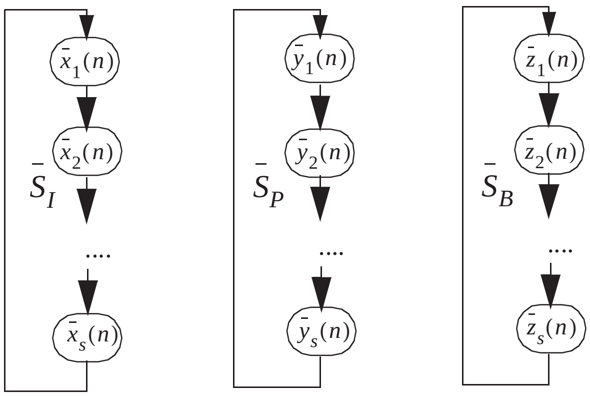

FIGURE 5. Periodic Markov Chain structure for each frame type used to modulate LPC slice process.

$S_I=\{x_1(n),x_2(n),...x_s(n)\}$, $S_P=\{y_1(n),y_2(n),...,y_s(n)\}$, and $S_B=\{z_1(n),z_2(n),...,z_s(n)\}$ where each set contains the sequence samples for specific slice positions within a particular frame type and s is defined as the number of slices within a frame. The mean is removed from the original sequences where: $S_I=\{x_1(n)-\mu_{x1},x_2(n)-\mu_{x2},...x_s(n)-\mu_{xs}\}$, $S_P=\{y_1(n)-\mu_{y1},y_2(n)-\mu_{y2},...,y_s(n)-\mu_{y2}\}$, and $S_B=\{z_1(n)-\mu_{z1},z_2(n)-\mu_{z2},...,z_s(n)-\mu_{zs}\}$ are model sets for the zero mean process. The zero mean process for each sequence is defined as,

$$\begin{aligned} x_i(n) &= a_{i,I}x_i(n-1)+G_{i,I}v(n) \\ y_i(n) &= a_{i,P}y_i(n-1)+G_{i,P}v(n) \\ z_i(n) &= a_{i,B}z_i(n-1)+G_{i,B}v(n) \end{aligned} \tag{4}$$

To initialize the autoregressive functions, we used the actual slice values, with means subtracted, from the first I, P and B frames. To get the final sample values, we add the means back into (4) to get

$$\begin{aligned} \bar{x}_i(n) &= x_i(n)+\mu_{i,I} \\ \bar{y}_i(n) &= y_i(n)+\mu_{i,P} \\ \bar{z}_i(n) &= z_i(n)+\mu_{i,B} \end{aligned} \tag{5}$$

Each sequence within a set, shown in Figure 5, is determined by the current state of the Markov Chain. The model is sufficiently modular in that if the video sequence does not contain B frames then the corresponding chain is removed. The selection of each chain is driven by a frame sequence state machine controlled by the MPEG encoding parameters N and M.

V. DESCRIPTION OF VIDEO SEQUENCES

The video sequences were encoded at a resolution of 352 x 240 pixels per frame, a frame rate of 30 frames/sec, and 15 slices/frame. The first three sequences: *Bike*, *Flowg*, and *Tennis* are 150 frames long, while "*Undersiege*" consisted of 731 frames. The encoding parameters M and N were set to 3 and 6 respectively. A slice consisted of one macroblock row of 352 x 16 pixels.

Bike is a sequence from the Corolco Pictures movie "*Terminator II*". It begins by showing a distant view of man on a motorcycle jumping from a platform. As the sequence continues, the motorcycle gets closer to the camera as it progresses down the screen. Towards the end of the sequence, the motorcycle is closest to the camera with a final side-view close-up of the rider.

The *Flowg* sequence shows a flower garden located in the bottom half of the screen and a row of houses in the background towards the top. The camera tracks this scenery from left to right.

Tennis is a sequence showing two men playing table tennis while a woman watches. It begins with a close-up of the ping-pong ball which is being bounced on a paddle by one of the players. The camera zooms out as one of the players serves. As the camera zooms out, the woman comes into view. There is a scene change where the camera focuses on the other player.

"*Undersiege*" is a sequence from the movie by Warner Brothers. It is a night scene which contains a lot of action with the actors firing weapons aboard a naval ship. The final portion of the sequence shows a spectacular shot of a helicopter blowing up and falling off of the deck of the ship.

A. Statistical Characterization

The trace file for each of the four sequences was created by using a parsing program written in C++ which extracted slice data from the MPEG elementary video streams. These traces were then used to statistically characterize each of the four video sequences. Some of these results are listed in Table 1. Since each video had the same GOP pattern and slice format, we could easily compare the effect of content on the bit-rate process. Also, since each slice covered one horizontal macroblock row of a picture, we can easily relate changes in the location of an object within a frame to the cells/slice plots.

Video	I	II	III	IV
Bike	1.23	19.61	6.44	6.21
Flowg	5.08	4.48	26.61	3.95
Tennis	2.29	10.20	12.02	4.41
UnderSiege	0.85	29.41	4.46	8.97

TABLE 1. Video statistics. (I) Avg bit rate 10^6 bits/sec, (II) Compression Ratio in pixels/bit, (III) Avg number of cells/slice, (IV) Cells per Slice peak-to-mean ratio.

We can see in Table 1 that of the four sequences *Flowg* had the highest mean bit rate, lowest compression ratio, and lowest peak-to-mean ratio for the number of cells/slice. This

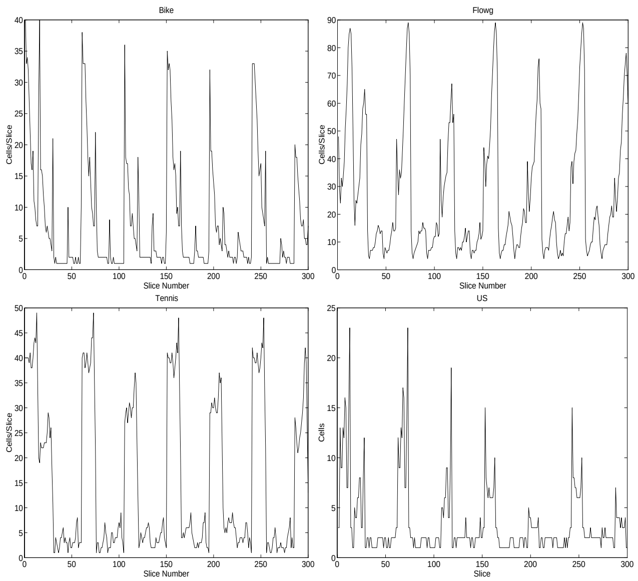

FIGURE 6. Plots of the number of ATM cells per slice for the first 20 frames of each video sequence.

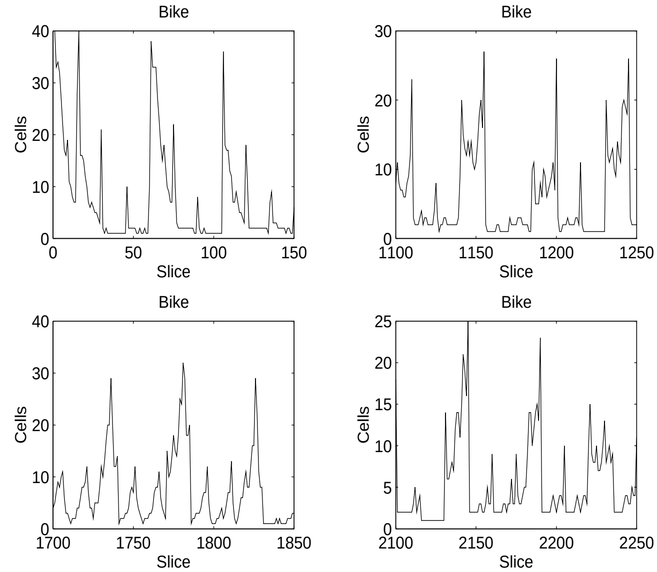

FIGURE 7. Cells/Slice plot of four different portions of th Bike Video.

was due to the many small flowers located in the flower garden. In contrast, "*UnderSiege*" has the lowest mean bit rate, highest compression ratio, and highest peak-to-mean ratio. This is a sequence which contains a lot of action which many would conclude would generate high bit-rate. To the contrary, the fact that this is a night scene indicates that the background of scene might be more significant in determinign overall bit-rate than any foreground motion. One might have expected *Flowg* to have a lower bit-rate than "*Undersiege*." One explanation for the difference is that apparently DCT does quite well in compressing a night scene, but does not do well with scenes containing many small objects. This implies that the characteristics of content, rather than motion activity, plays a more significant part in MPEG's effectiveness in compressing video sequences.

The statistical data also implies that there is a proportional relationship between compression ratio and the cells/slice peak-to-mean ratio (PMR). We see that as PMR increases the compression ratio also increases. We surmise that this is caused by the effectiveness of DCT as compared to motion compensation. For example, if a scene has consistent content over time then motion compensation will be very effective in finding appropriate motion vectors. This reduces the size *P* and *B* frames, thereby increasing PMR by reducing the mean bit-rate. Conversely, if motion compensation is not as effective, as in *Flowg*, the mean bit-rate increases which reduces PMR.

B. Slice Characterization Based on Content

We can see in Figure 6 that the number of cells per slice can differ significantly between videos. All of the sequences show distinctive pulses occurring at deterministic time intervals determined by the GOP pattern. Every forty-five slices there are alternating pulses caused by *I* and *P* frames. The GOP pattern for the sequences used in this study is IBBPBBI... Note that at the start of each sequence an *I* frame is followed by a *P* frame. This is required by the decoder in order to decode the first *B* frame. As a result, one can see that the first thirty slices are a concatenation of two pulses caused by an *I* and a *P* frame.

The spacing between pulses is due to the number of *B* frames between I/P frames. The pulse width is determined by the number of slices in a frame. One can see that the shape of the pulse differs significantly between videos. *Bike* shows pulses with peaks on the left side (first slice of the frame at the top of the picture) while *Flowg* shows pulses with peaks on the right side (last slice of the frame at the bottom of the picture). *Tennis* produces pulses which are relatively flat, almost resembling a square wave pulse. "*Undersiege"* generates pulses which are rather noisy. Since the encoding parameters and slice formats are the same for each video, pulse shapes must be due to the content. We surmise that the pulse shape is determined by the location of significant foreground objects within a picture. Some examples are: (1) The *Bike* sequence has a motorcycle in the upper right corner of the screen (left peak), (2) *Flowg* has the flower garden at the bottom of the screen (right peak), and (3) *Tennis* shows a close-up of a ping-pong ball being bounced on a paddle by one of the players (flat peak).

The shape of the pulse can change as the sequence progresses. This is shown in Figure 7 which shows a snapshot of the first, middle and last parts of the *Bike* sequence. We see that the peak moves from the left to right which is caused by the motorcycle moving from the top of the screen to the bottom as time progresses. During slices 1100 to 1250 we can see that the pulses get flatter, similar to that in *Tennis*.

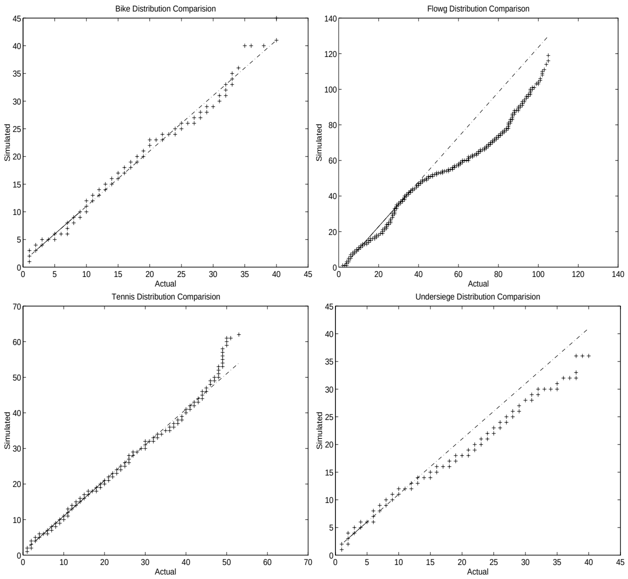

FIGURE 8. Distribution comparisons using QQ plots.

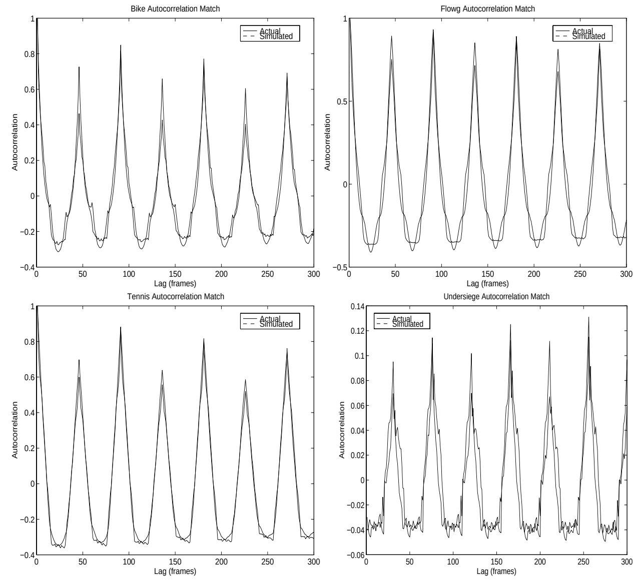

FIGURE 10. Autocorrelation comparisons.

This happens to be when the motorcycle is closest to camera and covers the majority of the screen.

Each of these videos will produce very different delay statistics, based not only on the content characteristics (night scene, foreground object, etc.), but on how much and how fast content changes over time (motorcycle moving from top-to-bottom of picture). It would be desirable to have a model which can capture these characteristics.

VI. SLICE MODEL RESULTS

We used the model outlined in section IV to determined if the statistical characteristics of the model compared well with the actual data. This was done by comparing the distribution and autocorrelation functions. We can see in Figure 8 that the QQ plots for each of the sequences is approximately linear indicating a good fit. *Flowg* does show a significant deviation, however, from the reference line as the number of cells/slice exceeds forty.

The autocorrelation function produced by the slice model, shown in Figure 10, is pseudo-periodic and compares quite well with the autocorrelation of the actual sequence. Matching the autocorrelation function is important since its behavior can have a significant impact on the performance of queueing systems [16].

The model was also used to compare simulation results of cell delay. We used the system shown in Figure 9 for an MPEG encoder producing slice data at deterministic intervals equal to τ / s, where τ is the frame period and s is the number of slices per frame. Each slice is then packetized and segmented into 53 bytes ATM cells. The cell payload equaled 47 bytes with an overhead of 6 bytes. The cells are then sent to an output buffer which is drained at a constant rate.

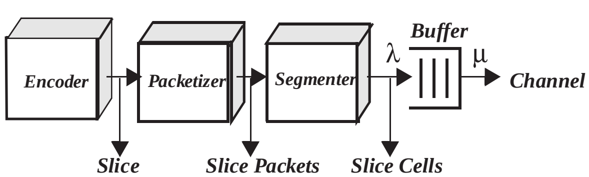

FIGURE 9. Encoding system with slice departures.

We compared the measured cell delay variation for offered loads (ρ) between 0.2 and 0.9. Cell delay variation is measured as the ratio of peak and mean delay. The simulations showed that both *Flowg* and *Tennis* compared well with the actual data for ρ up to 0.8 (we did not go beyond 0.8) while *Bike* and "*Undersiege*" compared reasonably well for ρ less than 0.3 and 0.6 respectively.

VII. CONCLUSIONS AND FUTURE WORK

We have presented a video model for the MPEG slice layer based on Linear Predictive Coding and the Generalized Periodic Markov Chain. The model is sufficiently modular in that it can be used for video sources which exclude *B* frames. The model produced an autocorrelation function which was pseudo-periodic and which matches the actual video trace data well. Results for the peak-to-mean ratio of cell delay matched the actual data well, especially for the *Bike, Flowg* and *Tennis* video sequences. The results for "*Undersiege*" deviated significantly when the traffic intensity exceeded 0.6 which is possibly be due to the longer length of this sequence. One drawback to the model is that each scene would require a different set of AR coefficients. However, a clustering algorithm similar to that used in [17] could be used in order to reduce the number of sets.

VIII. ACKNOWLEDGMENTS

We would like to thank the Portable Video Research Group for making the four video sequences available. We would also like to thank the University of California at Berkeley for providing the *mpeg_play* software decoder. Finally, we would like to thank the following individuals for their advice and comments regarding the subject of video modeling: Steve Blake, Mark Garrett, Benjamin Melamed, Dimitris Penderakis, and Olen Stokes.

IX. REFERENCES

1. CCITT Study Group XV, *Recommendation H.261 - Video codec for audiovisual services at p x 64 kbit/sec*, CDM XV-R37-E Recommendations of the H series., International Telegraph and Telephone Consultative Committee (CCITT), August 1990.
2. J. Enssle, "Modeling of Short and Long Term Properties of VBR MPEG Compressed Video in ATM Networks," *Proceedings of the 1995 Silicon Valley Networking Conference & Exposition*, San Jose, Ca. April 5-7,1995, pp 95-107.
3. M.W. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," *ACM SIGCOMM* 1994, pp 269-280.
4. D. Geist and B. Melamed, "TEStool: An Environment for Visual Interactive Modeling of Autocorrelated Traffic," *IEEE International Conference on Communications*, Vol. 2, pp. 1285-1289, June 1992.
5. D. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," *IEEE J. Select. Areas Commun.*, Vol. 2, No. 1, March 1992, pp. 49-59.
6. Changcheng Huang, Michael Devetsikiotis, Ioannis Lambadaris, and A. Roger Kaye, "Modeling and Simulation of Self-Similar Variable Bit Rate Compressed Video: A Unified Approach," *ACM SIGCOMM* 1995, pp.114-125.
7. ISO/IEC JTC 1 SC29, *Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s - Part 2: Video*, ISO/IEC 11172-2:1993(E), First edition, 1993.
8. *ISO/IEC JTC 1 SC 29 WG 11, Information Technology - GENERIC CODING OF MOVING PICTURES AND ASSOCIATED AUDIO INFORMATION: VIDEO*, Recommendation H.262, ISO/IEC 13818-2, Draft International Standard, May 10, 1994.
9. B. Jabbari, F. Yegenoglu, Y. Kuo, S. Zafar, and Y. Q. Zhang, "Statistical Characterization and Block-Based Modeling of Motion-Adaptive Coded Video," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 3, No. 3, June 1993, pp 199-207.
10. P.R. Jelenkovic and B. Melamed, "Automated TES Modeling of Compressed Video," *IEEE INFOCOM*, Boston, Massachusetts, pp 746-752, April 1995.
11. M. Kawarasaki and B. Jabbari, "B-ISDN Architecture and Protocol," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 9, December 1991, pp. 1405-1415.
12. F. Kishino, K. Manabe, Y. Hayashi, and H. Yasuda, "VBR Coding of Video Signals for ATM Networks," *IEEE J. Select. Areas Commun.*, Vol. 7, No. 5, June 1989, pp. 801-806.
13. R. Landry and Ioannis Stavrakakis, "Multiplexing Generalized Periodic Markovian Sources with an Application to the Study of VBR Video." *Intl. Conf. on Comm. 1994*, Vol. 2, pp. 1014-1018.
14. A. Lazar, G. Pacifini and D. E. Pendarakis, "Modeling video Sources for Real-Time Scheduling," *IEEE Globecom 1993* Vol. 2, pp. 835-839
15. D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Commun. of the ACM*, Vol. 34, No. 4, April 1991, pp 47-58.
16. M. Livny, B. Melamed, and A. K. Tsiolis, "The Impact of AutoCorrelation on Queueing Systems," *Management Science*, Vol. 39, No. 3, March 1993, pp. 322-339.
17. B. Melamed and D. Penarakis, "A TES-Based Model for compressed *Star Wars* Video," *IEEE GLOBECOM 1994*, pp. 120-126.
18. B. Melamed, D. Raychaudhuri, B. Sengupta, and J. Zdepski, "TES-Based Video Source Modeling for Performance Evaluation of Integrated Networks," IEEE Trans. on Comm., Vol. 42, No. 10, pp. 2773-2777, October 1994.
19. J. Proakis, *Digital Communications*, Second Edition, McGraw-Hill, New York, 1989.
20. W. Verbiest, L. Pinno, and B. Voeten, "The Impact of the ATM Concept on Video Coding," *IEEE J. Select. Areas Commun.*, Vol. 6, No. 9, December 1988, pp. 1623-1632.
21. F. Yegenoglu, B. Jabbari, Y. Q. Zhang, "Motion-Classified Autoregressive Modeling of Variable Bit Rate Video," *IEEE Trans. on Circuits and Systems for Video Tech.*, Vol. 3, No. 1, February 1993, pp 42-53.
22. J. Zdepski, K. Joseph, and D. Raychaudhuri, "Packet Transport of VBR Interframe DCT Compressed Video on a CSMA/CD LAN," *IEEE GLOBECOM 1989*, pp. 886-890.