Pseudo-Modeling of MPEG-Based Variable-Bit-Rate Video Streams

Wu-chi Feng
Dept. of Comp. and Info. Sci.
The Ohio State University
Columbus, OH 43210
wuchi@cis.ohio-state.edu

Abstract: *In this paper, we introduce a novel technique for the modeling of variable-bit-rate MPEG video streams. This technique is designed to aid multimedia systems researchers interested in using real compressed video data to study the systems they are designing without the overhead of having to digitize and compress the video stream multiple times. Our proposed approach differs from current approaches in that is uses an inexpensive non-inter-coded video capture board to digitize the movie data. Once captured, our model uses a subsample from the digitized data to create a linear regression model of the target video (both quality and frame pattern). Using this linear regression model, our approach statistically generates the target video. We have digitized and compressed 26 hours of constant-quality MPEG video to verify our pseudo-modeling approach.*

Keywords: MPEG, video traffic modeling, video streaming

1. Introduction

Mathematical modeling of variable-bit-rate video streams such as MPEG-1 or MPEG-2 video streams is useful in the research and design of multimedia systems. Uses of these models include the design of video server structures, quality-of-service architecture design, multimedia networking issues, and low-level networking queue management policies. By modeling these video streams, researchers can investigate the design space of their systems without having the large overhead of the implementation of an actual system. These models attempt to capture the multiple time-scale characteristics that video compression algorithms introduce. Short-term characteristics are determined by the inter-coding of video frames (i.e. the frame pattern such as found in MPEG). Long-term characteristics include the changes in scenes that occur over the duration of the video. The use of analytically derived video sequences, however, leaves the designer wondering how their approach will work when an actual video stream is used.

On the other hand, capturing and compressing large amounts of real video data can be cumbersome, expensive and time-consuming (particularly predictive coded video such as MPEG). While it is possible to capture MPEG video sequences in hardware and in real-time, these systems are often expensive (on the order of $40K) and typically compromise on the motion-compensation algorithms in order to achieve real-time compression. In addition, if the researcher is interested in the effect of different encodings of a particular video sequence such as a change in the frame pattern or the quantization levels (for constant-quality video encoding), the same video sequence will have to be re-captured and re-compressed for as many different types of configurations that need to be explored.

In this paper, we introduce a novel technique for the "pseudo-modeling" of MPEG video sequences for the research and design of multimedia systems, particularly research based on simulation. This work is motivated by several factors. First, basic video capture boards that are capable of only intra-coded video capture are relatively inexpensive (on the order of several hundred dollars) and result in fairly high quality video streams. These boards include uncompressed video capture boards, I-frame only MPEG capture boards, and Motion-JPEG capture boards; Manufacturers of these boards include companies like Osprey and Targa; Broadway; and MiroVideo, respectively. Second, after capturing a single video stream, it would be nice to be able to model *inter-frame* MPEG video streams from an *intra-frame* only coded video stream. This would allow the researcher to re-use a *single* compressed video stream for the exploration of *multiple* types of video sequences, including different frame patterns and frame qualities.

At a high-level, our proposed pseudo-modeling approach takes a single Motion-JPEG or all I-frame MPEG compressed video sequence and models more complex MPEG video sequences with P and B frames through statistical modeling of the video stream. This simple approach will allow researchers to (i) capture the larger I-frame data and the long-range dependence that these video sequences exhibit and (ii) allow the smaller predictive coded frames to be statistically generated for the simulations.

In the following section, we briefly review some of the background material necessary for this paper and highlight some of the research efforts in modeling variable-bit-rate video streams. In Section 3, we present our pseudo-model for MPEG video streams. Using nearly ***26 hours*** of MPEG compressed video sequences (~2.8 million compressed video frames), we describe some of the statistical properties of MPEG video that we have observed in Section 4. In Section 5, we describe the experiments that we conducted to test our model. Finally, we summarize our findings in Section 7.

Contributions of This Work: This work makes two main contributions to multimedia systems. First, this work is based on a significant amount of MPEG compressed video data. We believe that this is one of the first works that takes an in-depth look at the effect of MPEG compression for long video sequences. We expect to distribute our statistical information (i.e. frame sizes) from these MPEG sequences through the final ACM Multimedia 1999 web site for other researchers to use (or have a pointer to our web site if disk space is a problem). Second, and more importantly, this work introduces a novel MPEG modeling technique based on a single captured video sequence. This work bridges mathematically modeling video streams and strictly using captured video sequences for simulation research.

2. Background and Related Work

In this section, we briefly describe some of the background and related work necessary for this paper. In particular, we briefly highlight the MPEG video compression standard and the related work in variable-bit-rate video modeling

2.1. MPEG Video Compression

The MPEG video compression standard has a target compression rate of 1.5 Mbps for a video stream at 352x240 pixel resolution and its related audio signal. The MPEG video compression algorithm is a lossy DCT-based compression standard consisting of three distinct frame types: *I, P,* and *B* frames. The *I* frames are independently coded (intra-coded). As a result, I-frames are very similar to those frames in a Motion-JPEG sequence. *P*-frames are predictive coded from a previous *I* or *P* frame, taking advantage of the similarities between frames in a video sequence. Finally, *B*-frames are bi-directionally predictive coded from both a future and past frame. Thus, *B*-frames offer the highest degree of compression at the added expense of a large motion compensation overhead.

In the encoding of MPEG-video sequences, a frame pattern is typically chosen such as

I B B P B B I B B P B B I ...

and is used to drive the compression. While the compression standard itself does not force this repeated pattern (virtually any frame sequence is allowed), using a fixed pattern makes the implementation of the video encoder simpler. Using the frame pattern in the compression of the video sequence, the resultant stream exhibits burstiness at multiple time scales. The frame pattern itself causes short-term burstiness. The variation in scene content causes burstiness at a much larger time scale. This long-term burstiness is sometimes referred to as the *long-range dependence*[11]. Readers interested in a more detailed description of MPEG are referred to one of its overview papers[10].

2.2. Related Work

Several models have been proposed in the literature for the modeling of variable-bit-rate video streams such as MPEG. Many of these techniques observe (and we agree) that the long-term variation in compressed video sequences are due to the variation in scene content and that the short-term burstiness is due to pattern burstiness of the video compression standard being used. As a result, much of the modeling of MPEG video sequences focuses on producing scene level variations.

In [9], Lazar, Pacifici, and Pendarakis propose a model that models video sequences as a collection of scenes. Within a scene, a statistical model is developed for both the sequences of frames and the slices within the frames. In [8], the authors introduce a model based on M/G/∞ input processes. The authors evaluated this model primarily on the autocorrelation function of the frames within the video sequence. The main focus of this work is the capture of the long-range dependence that video streams exhibit. [1]

3. The Video Model

Our proposed video model is based on the assumption that we would like to use actual compressed video traces to drive simulations for research and development. However, we do not want the large overhead involved with re-coding the *same* video stream to test different configurations of the bit stream. At a high-level, our pseudo-modeling approach takes as input a *single* MPEG-1 (all I-frame encoded) or Motion-JPEG compressed video stream and then generates an MPEG-1 video sequence with inter-coded *P* an *B* frames based on a subsample of the video stream. The entire process is outlined in Figure 1. We will describe this process in more detail in the following paragraphs.

To pseudo-model video frames within the stream, we first select a representative sub-region (typically less than 5 minutes will suffice) within the original video sequence and then re-code it into the “target” video format. The “target” format includes both the frame quantization levels and frame pattern type. Selection of the sub-region will be discussed in the experimentation section. Using the resultant re-coded sub-region, we then split the stream into the various frame types: *I*, *P*, and *B* for MPEG video sequences. For each class of video frames, we plot the pairs <original coded video frame size, re-coded video frame size> for all the frames that were re-coded. We use linear regression analysis per frame type to create a “best-fit” line that minimizes the total mean square error. Details of linear regression analysis can be found in [6]. The linear regression for each frame type results in three parameters:

- *m* - the slope of the linear regression line
- *b* - the *y*-intercept of the linear regression
- *std_err* - the resulting standard error in the estimation

Using these parameters, we then generate a pseudo-modeled video sequence from the original all I-frame MPEG video sequence or the Motion-JPEG video sequence. (Again, we reiterate that we assume the video stream has been constant-

---

[1] Note to reviewers: There is more related work than we were able to put in this page and font limited document. We would appreciate some suggestions on what areas to remove if you feel that more related work is necessary.

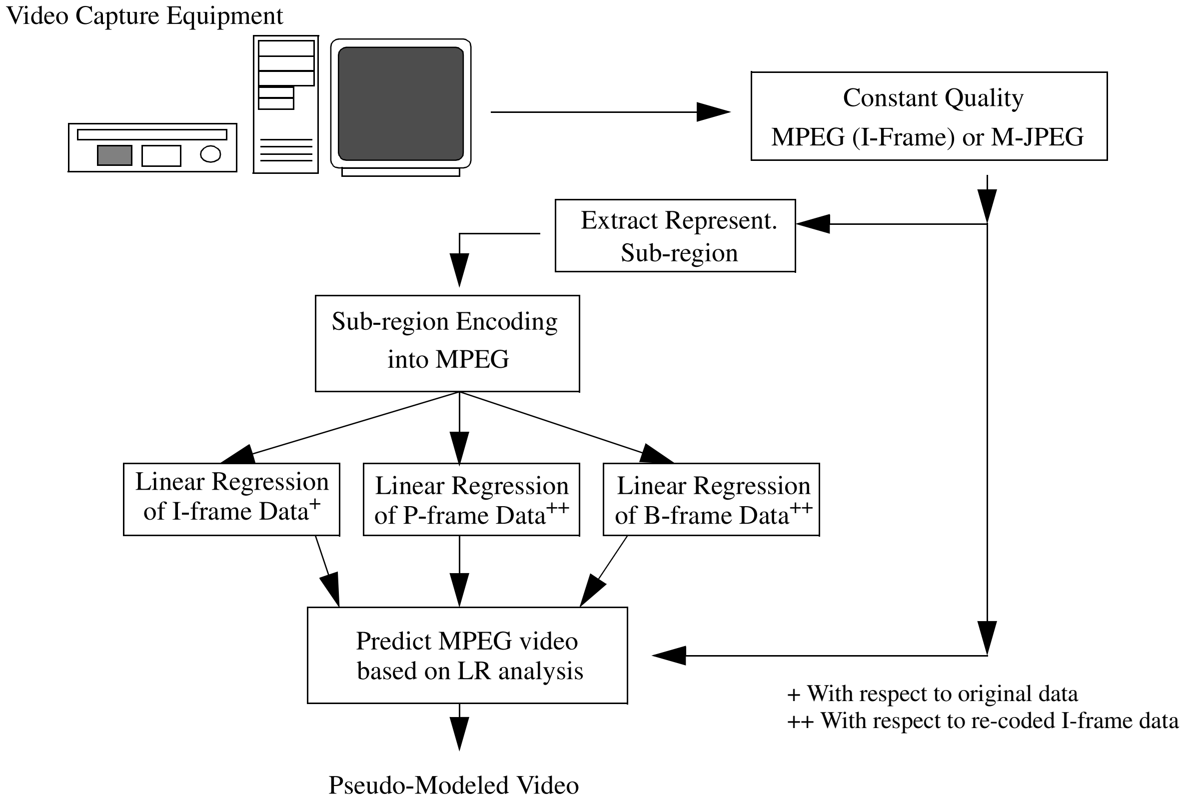


Figure 1: Pseudo-Modeling Architecture: This figure shows the pseudo-modeling architecture. The input into this system is an MPEG (all I-frame encoding) or M-JPEG video stream that can be easily and inexpensively captured. The output of the system is a sequence of video frame sizes that are predicted based on a sub-sample encoding and linear regression analysis.

quality compressed). Let each frame of the original sequence be denoted as f_j. The pseudo-modeled video frame is the defined as:

$$f_j' = f_j*m + b + \text{pertabation}$$

where the pertabation is derived from a normally distributed random number generator centered around 0 width a sigma of *std_err*. The pertabation component simply provides some "randomness" in the frame distribution that exists in the original video sequence.

In the modeling of the video stream, there are several small implementation details that we had to deal with. First, it is possible that the evaluation of the $f_j*m + b$ component can result in a *negative* number. This occurs when the original frame size is fairly small with respect to the linear regression line for the frame type. In this case, we replace b with $(b/2)$ in the model equation for f_j'. Second, the calculation of the pertabation may also result in a large negative number. We limit the effects of very large pertabations (no matter how unlikely), by making sure the normally distributed pertabation lies within 2 sigmas of the mean pertabation (0). In either of the above cases a *negative* frame size is still possible. In the unlikely event that this occurs, we set the generated video frame to a minimum value of 28 bytes (which we have observed in MPEG-1 video compression). Note, we also set any pseudo-modeled video frame less than this minimum to 28 bytes as well.

The accuracy of the pseudo-model will obviously depend on the selection and size of the subregion selected. Too small a sub-region may result

Name	Pattern	I,P,B Quant.	Psearch/Bsearch	Total Frames	Compressed bit rate (Mbps)
CD_122	IBBPBBI...	1, 2, 2	Log. / Cross2	174529	2.27
CD_244	IBBPBBI...	2, 4, 4	Log. / Cross2	174529	1.11
CD_8	IIIIIII...	8, -, -	-- / --	174529	1.38
CD_81025	IBBPBBI...	8, 10, 25	Log. / Cross2	174529	0.31
ET_122	IBBPBBI...	1, 2, 2	Log. / Cross2	204679	1.55
ET_2	IIIIIII...	2, -, -	-- / --	204679	3.09
ET_244	IBBPBBI...	2, 4, 4	Log. / Cross2	204679	0.75
ET_244	IBBPBBI...	2, 4, 4	Subsample / Simple	204679	0.76
ET_8	IIIIIII...	8, -, -	-- / --	204679	1.06
ET_810	IPPPPPI...	8, 10, -	Log. / --	204679	0.25
ET_81216	IBBPBBI...	8, 12, 16	Log. / Cross2	204679	0.23
ET_81025_LC	IBBPBBI...	8, 10, 25	Log. / Cross2	204679	0.23
ET_81025_SC	IBBPBBI...	8, 10, 25	Subsample / Cross2	204679	0.23
ET_81025_SS	IBBPBBI...	8, 10, 25	Subsample / Simple	204679	0.23
ET_81025_EE	IBBPBBI...	8, 10, 24	Exhaust. / Exhaust.	22,734	0.19

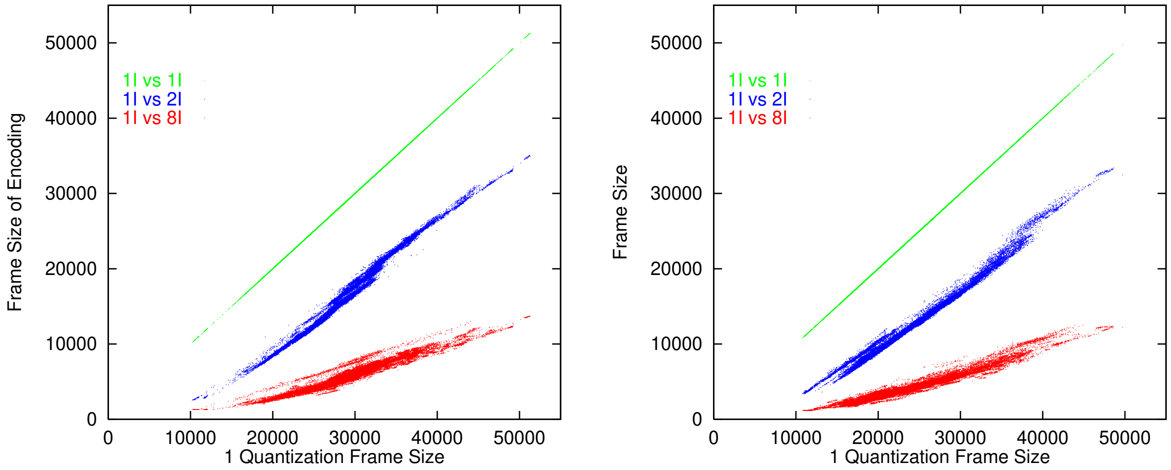
Figure 2: Video Statistics Table. This table lists the relevant statistics for the video sequences that digitized and compressed into MPEG-1 format. All of these streams were compressed at 352x240 pixels at 30 frames per second.

in an inaccurate model, while too large a subregion may result in additional overhead to the research (something to definitely be avoided!). As we will discuss later, we chose a small sub-region (5-6 minutes) in the linear regression analysis for the entire video stream to better approximate what researchers may be willing to do for the pseudomodel.

4. Compressed Video Analysis

To bring some validity to our approach, we have captured and compressed two movies *E.T. - the Extra Terrestrial* and *Crocodile Dundee* into several MPEG-1 compressed formats. We captured this video data using a Pioneer Laser Disc player and a MiroVideo DC 30 Plus video capture board. The MiroVideo DC 30 Plus board digitizes the data into the Motion-JPEG format. We then used the Berkeley MPEG encoder (version 2.0) to compress the actual video data. The statistics for the resulting streams are shown in Figure 2. In total, we compressed approximately *26 hours* of MPEG-1 video frames representing approximately *2.9 million* compressed video frames.

We chose the various parameters to get a variety of encodings to understand some of the tradeoffs that the encoder makes. All video sequences are compressed with a constant quantization level for the I, P, and B frame coefficients. The default quantization levels for the encoder are 8, 10, and 25 for the I, P, and B frames, respectively. Using these settings results in a fairly small encoded bitrate for our video sequences. We believe that this is probably due to the wide screen format that the laser disc format adopts to maintain the original cinema aspect ratio. That is, these video streams were in the *wide screen* format resulting in a black stripe across the top and bottom of the video



a) *Crocodile Dundee* b) *E.T.*

Figure 3: I-frame MPEG Analysis. These figures show the x-y scatter plot for the I-frames in the video sequences encoded with quantization levels (1,2,2), (2,4,4), and (8,10,25) with respect to the (1,2,2) encoding. Thus, point (22000, 10000) represents an I-frame that requires 22,000 bytes to represent at a quantization level of 1 and that requires 10,000 bytes at a quantization level of 2.

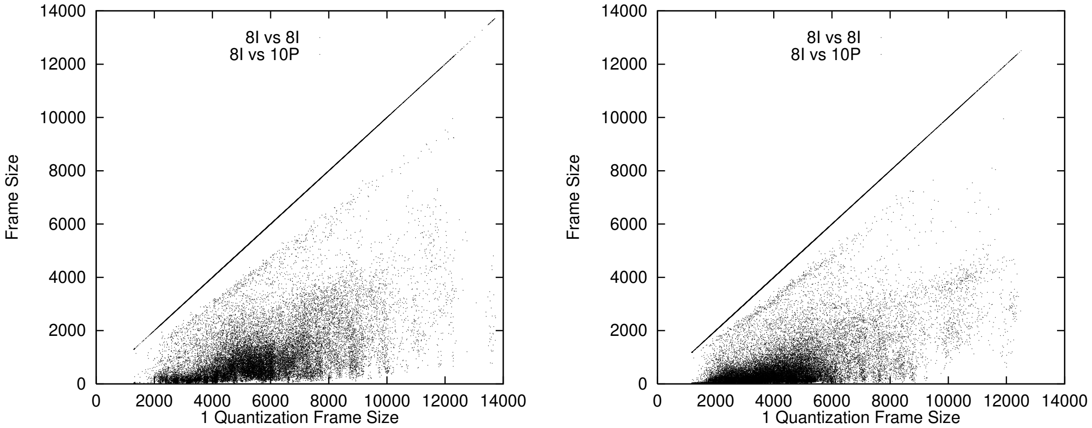
screen. For the motion-compensation search, we had a variety of options available to us for both the P and B frame video searches. The *Logarithmic* and *Cross2* search algorithms take a subsample of points as possible motion search matches and then recursively searches again with a smaller region at the point that had the best match. A description of these search algorithms can be found in [2].

Out of curiosity, we were interested in how much better the *Exhaustive* search algorithm is for the motion compensation algorithm instead of using one of the simpler search algorithms. While Figure 2 (last row) shows that the exhaustive search algorithm has the smallest bit-rate, this is actually somewhat misleading. Using a 400 MHz Xeon Pentium Processor with 512 MBytes, encoding the 22,734 frames required *30* days of straight computation. The encoded segment is from the start of side 2 on the laser disc (i.e. somewhere in the middle of the movie). Comparing this segment *with the same segment* in the other ET movies encoded with 8, 10, and 25 quantization levels for the I, P, and B frames lead us to a somewhat surprising result: *using the exhaustive search had no net gain in compression*. We found that many frames were actually larger in the exhaustively compressed video sequence (in addition to many smaller ones). We suspect that the exhaustive matching criteria resulted in slightly *more* coefficients in the DCT, causing a slightly higher bit rate in some circumstances. The motion compensation algorithm uses the mean absolute difference measurement [2].

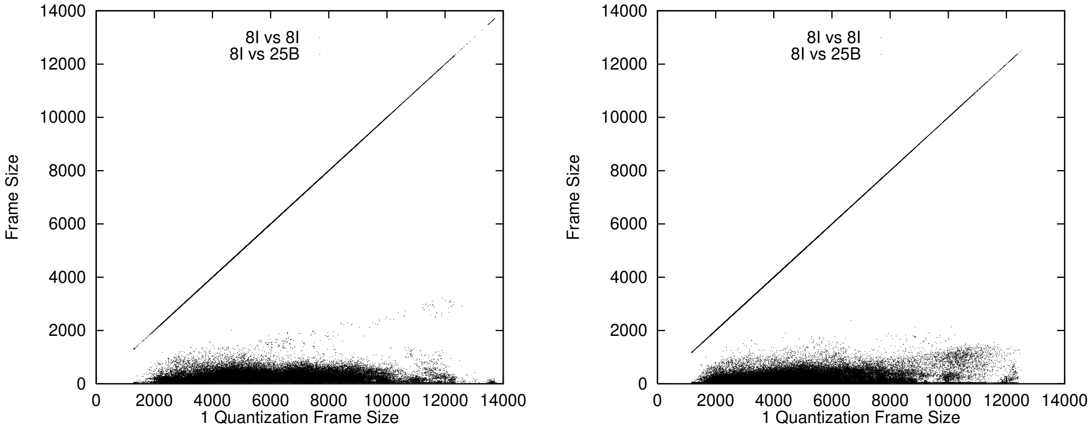
A Brief Analysis of the Compressed Video

In this subsection, we briefly highlight some of the insights that lead us to propose the pseudo-model approach. In Figure 3, we have graphed all the I-frame pairs of: (ET_122, ET_122), (ET_122, ET_244), and (ET_122, ET_81025). The straight line, y=x, are the (ET_122, ET_122) points. As shown by the figure, both movies exhibit a strong correlation between the coarser quantization level encodings and the I-frames from the ET_122 sequence. The correlation coefficient between the 1 quantization and the 2 and 8 quantization I-frames are greater than 0.94. Each graph in the figure represents approximately 30,000 frames of video.

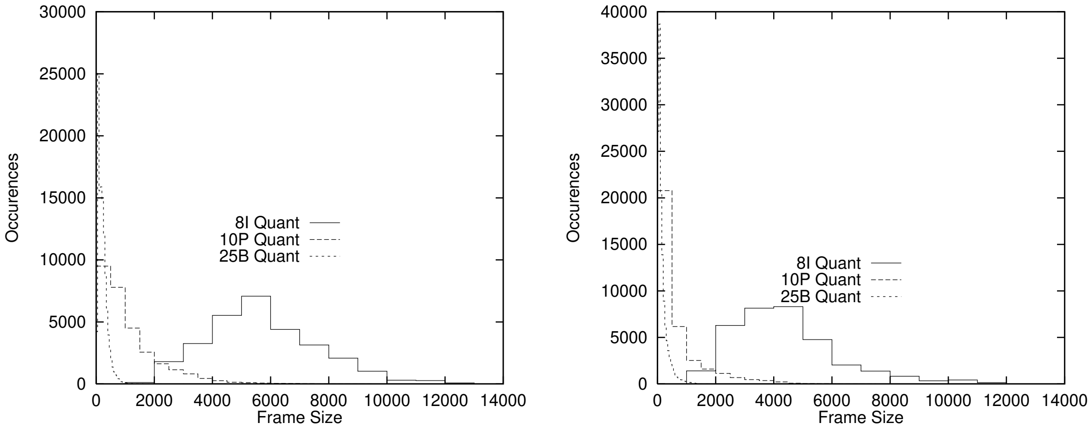
In Figure 4, we have graphed a similar analy-



a) *Crocodile Dundee* - P frame distribution b) *E.T* - P frame distribution.



c) *Crocodile Dundee* - B frame distribution d) *E.T* - B frame distribution



e) *Crocodile Dundee* - frame size histogram f) *E.T.* - frame size histogram

Figure 4: P and B Frame Analysis: This figure shows the distribution of P and B frames from ET_81025_LC compared to the I-frames in the sequence ET_8. Figures (e) and (f) shows the histogram of the distribution of all frames in the ET_81025_LC video sequence.

sis of the *P* and *B* frames within the MPEG video sequence. In figures (a) and (b), we see that there is a large density of video frames in the lower left hand corner. These frames are typically the fade in and fade out frames that compress well (and easily). Toward the right and upper side of the graphs, there are two distinct regions of frames: a higher density of frames towards the right and a slightly smaller density of frames near the *y*=*x* line. The frames that tend to be closer to the *y*=*x* line are mainly due to the use of a fixed pattern in the video sequence. Here, the MPEG encoder is forced to use a particular encoding across a *scene* boundary. As a result, the *P* frame has a large number of intra-coded (*I*-frame) macroblocks because there is insufficient redundancy between the *P* frame and the past frame it is trying to be differenced from. We note that the *P*-frames shown above approximately 2000 bytes are less than 10% of all the *P* frames within the movie.

In figures (c) and (d), we see the advantage that bi-directionally encoded video frames have over *P* frames. In these figures, we see that the *B* frames have a much denser distribution of data. Because the MPEG encoder has many choices for compression (at the expense of more processing requirement), it is effective at finding an appropriate motion compensated block to use, either in the future or the past reference frame. That is, the motion compensation is allowing may of the blocks to be coded as skipped. Thus, the B-frames contain a large percentage of no DCT coefficients but just motion vectors. We also suspect that the larger *B* frames in the *Crocodile Dundee* video are due to very quick action scenes with large amounts of fast pans taking the video out of the motion search window.

5. Experimentation

In this section, we describe the experiments that we used to evaluate our pseudo-modeling approach. We first describe how we selected a sub-region for the modeling and then describe the performance of the modeling algorithm in two example application areas.

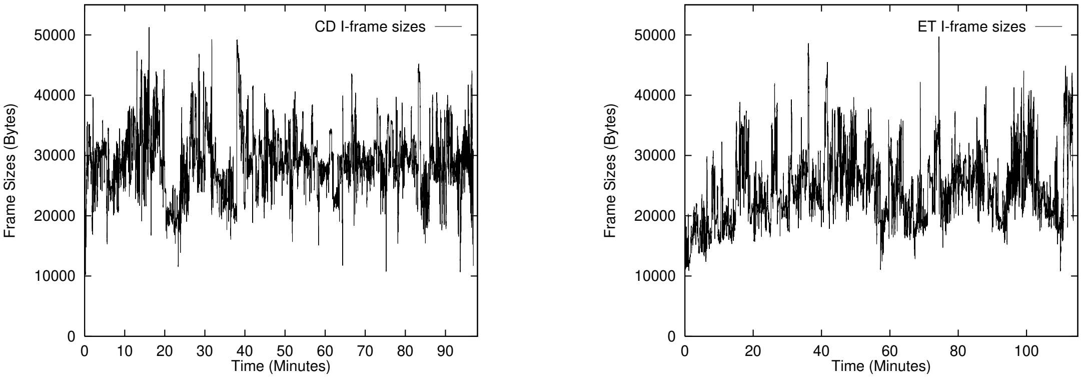
5.1. Pseudo-Model Sub-Region Selection

One of the important aspects of pseudo-modeling is that you *do not* want to encode the entire video stream to figure out what the model parameters should be (or you'd rather use the actual data!). The question then becomes how much of the video stream do you need to encode to get the approximation parameters necessary for pseudo-modeling. Intuitively, a sample from the video stream that captures a large portion of the frame sizes is the most useful. For the modeling, we picked two regions from the *E.T.* movie (minutes 36-37, and minutes 70-75). We first picked the 70-75 minute region because is seemed to contain the broadest variety of frame sizes. Figure 5 shows the I-frame data for the two movies that we have been examining. We see that the 70-75 minute region in *E.T.* has both small and large frame sizes in it. After examining this region, we found that there were actually very few frames in the region above 40,000 bytes so we added the 36-37 minute region to the linear regression sample, leaving a total of 6 minute out of the 99 minutes in the movie. The resulting I-frames (compared with the I-frames from the 1 Quantization frames) are shown in Figure 6(b). As shown in this figure, much of the basic shape in Figure 3(b) has been captured.

For the movie *Crocodile Dundee*, we selected the region from time 38-43 minutes, a five minute segment for the linear regression analysis. The sub-region I-frames compared with the I-frames from the 1 Quantization frames is shown in Figure 6(a). Again, we see that by choosing a region that covers a large portion of the frame size space allows the linear regression to be performed on points similar to the entire video stream being re-coded.

5.2. Applications of Pseudo-Modeling

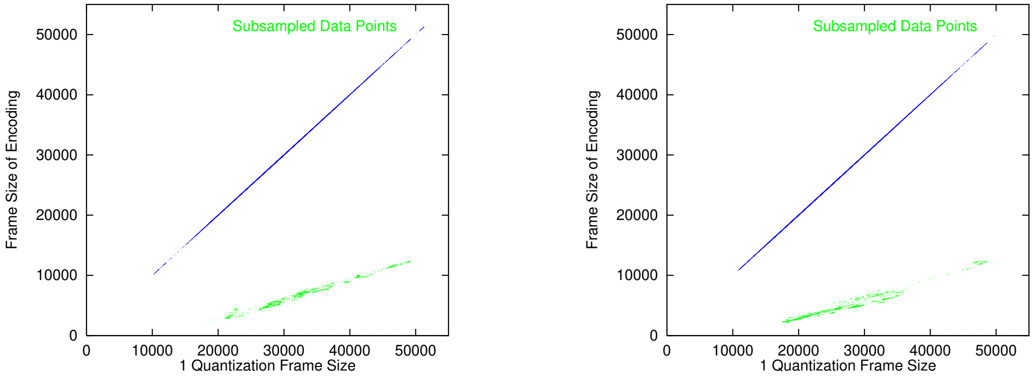
In this section we highlight the use of pseudo-modeling for MPEG videos in two application areas: autocorrelation for long-range dependence analysis and bandwidth smoothing techniques. Using the linear regression numbers determined in



a) *Crocodile Dundee*

b) *E.T.*

Figure 5: MPEG-video I-frame Data. This figure shows the I-frames encoded with quantization level 1 (i.e. the highest quality) from the CD_122 and ET_122 MPEG-1 video sequences.



a) *Crocodile Dundee*

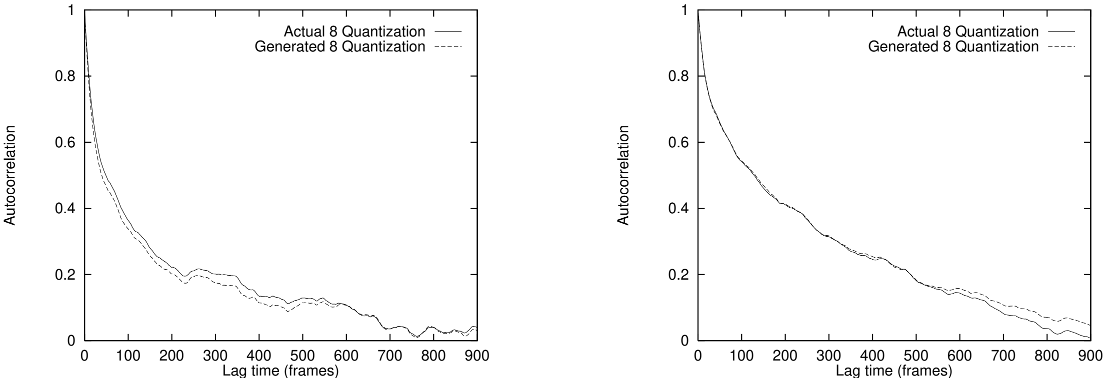
b) *E.T.*

Figure 6: Video Subsampling. These figures show the result of sub-region selection for the linear regression analysis. For the movie *Crocodile Dundee*, minutes 36-37 and 70-75 are shown in figure (a). For the movie *E.T.*, the 8 quantization vs. 1 quantization I-frames for minutes 38-43 are graphed in figure (b).

Section 5.1, we then applied this to the entire video sequence to create a "generated" MPEG-1 video sequence. We compare the performance of the generated video sequence with that of the actual video stream that it is trying to model.

Autocorrelation

The autocorrelation function describes the long-range dependence a video stream exhibits and has been the focus of several MPEG-1 video modeling papers. Typically, the autocorrelation function is used on the independent frames within the video stream [8] but has been used on inter-coded video sequences as well [9]. To test the performance of a pseudo-modeled video stream for this application, we used the linear regression model that was derived in Section 5.1. The results of the linear regression for *Crocodile Dundee* were (m = *0.338, b = -4241, std_err = 292*), while the results for the movie *E.T.* were (m = *0.311, b = -3520, std_err = 451*). It is interesting to note that the linear regression analysis is very similar to that of the entire coded sequence and both movies have very comparable linear regression parameters.



a) *Crocodile Dundee*   b) *E.T.*

Figure 7: Autocorrelation Analysis. These figures show the autocorrelation analysis using the actual video stream sequence and the pseudo-modeling approach for *Crocodile Dundee* and *E.T.*

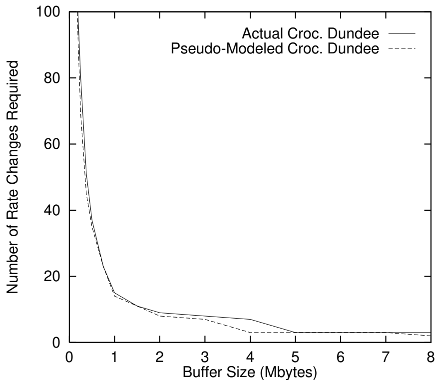
The results of the autocorrelation analysis are shown in Figure 7. As shown in the graphs, using the pseudo-model results in autocorrelation functions that are nearly identical to the original, even though we used just 5 and 6 minute samples for the linear regression analysis. While we merely guessed the region to use in the analysis, our first intuitive guess seemed to work well. Future work will consider the actual selection process for these sub-regions and their impact on the models.

Bandwidth Smoothing

Bandwidth smoothing techniques are used to efficiently allocate network and server resources for the continuous delivery of compressed prerecorded video streams [3]. By analyzing the *a priori* information available from stored video streams (i.e. the frame size information), the delivery of the video stream can be planned to optimize certain metrics. Using a client buffer to prefetch data, the server can send bursts of large frames *before* they happen so that during the burst, the client can augment network bandwidth with bandwidth from the buffer. The two bandwidth smoothing algorithms that we highlight here minimize the peak bandwidth requirements for the delivery of stored video streams while minimizing the total number of rate changes[4] or minimizing the variability of bandwidth requirements[12]. Finally, we note that these bandwidth smoothing algorithms assume that the client-side buffer has a fixed maximum size.

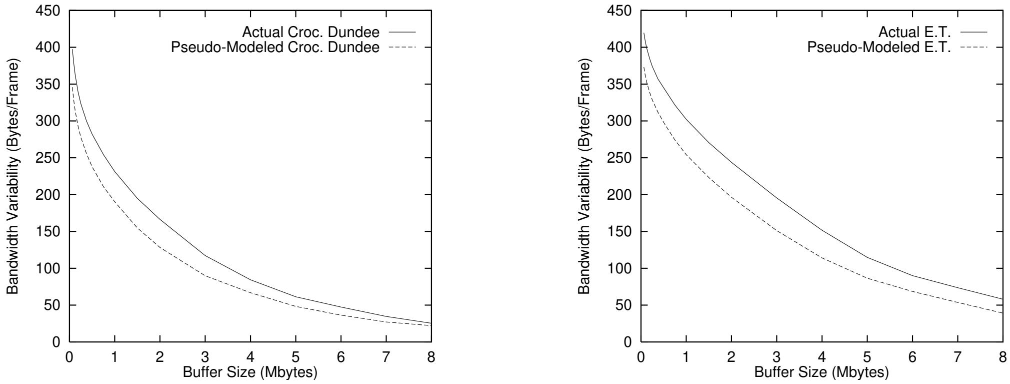
Our first experiment examines the effect of pseudo-modeling on the total number of rate changes for the *minimum changes bandwidth allocation algorithm*. Using the movie *CD_8* and *ET_8,* we used the same linear regression analysis as presented in Section 5.1. In this experiment, however, we modeled both the B and P frames in the video stream. For the *Crocodile Dundee* movie, we used the linear regression to transform the *CD_I8* movie into a pseudo-modeled *CD* movie with quantization 8, 10, 25. We then compared this with the actual *CD_81025* movie. The resulting performance for the actual and pseudo-modeled video streams are shown in Figure 8. Here, we see that the actual and pseudo-modeled approach result in very similar graphs even under small buffer requirements.

In Figure 9, we have graphed the bandwidth variability graphs for the *minimum variability bandwidth allocation algorithm*. Here, we see that the pseudo-modeled video stream results in a smaller bandwidth variability in the stream. We believe that this is due to the fact that by using a randomly generated pertabation that in some instance a large positive pertabation get canceled by a large negative pertabation in adjacent frames. Nevertheless, the graphs are comparable, typically differing by



a) *Crocodile Dundee* b) *E.T.*

Figure 8: Bandwidth Smoothing Changes. These graphs show the performance of pseudo-modeled MPEG video sequences (with frame pattern IBBPBBI...) from all I-frame encoded sequences for the movies *Crocodile Dundee* and *E.T.* Each graph shows for a given buffer size, the result of the *minimum changes bandwidth allocation algorithm.*



a) *Crocodile Dundee* b) *E.T.*

Figure 9: Bandwidth Smoothing Variability. These graphs show the performance of pseudo-modeled MPEG video sequences (with frame pattern IBBPBBI...) from all I-frame encoded sequences for the movies *Crocodile Dundee* and *E.T.* Each graph shows for a given buffer size, the result of the *minimum variability bandwidth allocation algorithm.*

less than 10%. We note that since the basic performance trends has been captured, the pseudo-model generated movie is most likely very similar to a movie with *slightly* different coding parameters. (For example, I, P, B frame quantization levels of 9, 12, and 27 instead of the 8, 10, 25 used here).

6. Acknowledgments

This work was supported in part by grants from the National Science Foundation, Ameritech, and the Ohio State Research Foundation. The work reported in this paper do not necessarily reflect the views of the funding agencies.

7. Conclusions and Future Work

In this paper, we have introduced a novel technique for the modeling of MPEG movie data based on a Motion-JPEG or all I-frame encoded MPEG video stream. This technique allows researchers to use real compressed video data to capture the right long-term dependence but without the overhead of having to re-code the video sequence if more than one quality level or frame pattern type are desired.

Our approach takes a small sub-region from an all I-frame MPEG sequence or a Motion-JPEG sequence, which can be easily and inexpensively captured with commodity components, and uses a sub-region to create a linear regression model of the video stream (quality and frame type) to be modeled.Our results show that the pseudo-model works fairly well at capturing the "essence" of the video streams with just a small sample (< 5% of the entire movie trace) without the large overhead in doing the entire compression.

We have captured and compressed *26* hours of compressed video data to study the pseudo-modeling approach. We have presented an in-depth analysis of this data and have found some interesting trends in MPEG encoders. We hope to make these streams available for multimedia systems researchers as part of the ACM Multimedia '99 web page.

Future work includes a more in-depth analysis of sub-region selection and its impact on the synthetically generated video stream. We are also in the process of capturing several more video streams to add to the library that we have produced to continue the verification of our results. We plan to explore more of the MPEG encoding parameters. In particular, the MPEG encoder from Berkeley has a maximum search window of +/- 10 pixels. We would like to investigate whether playing with this window results in a beneficial or detrimental effect on the compression. Finally, while the MPEG-2 and MPEG-1 video standard are very similar (for non-interlaced video), we are interested in extending this model for MPEG-2 video streams.

8. References

[1] J. Beran, R. Sherman, M. Taqqu, W. Willing, "Long-Range Dependence in Variable-Bit-Rate Video Traffic," *IEEE Transactions on Communications*, Vol. 43, pp. 1566-1579, 1995.

[2] V. DeKalb, *Video Processing,* McGraw-Hill Publishers, 1995.

[3] Wu-chi Feng, S. Sechrest, "Smoothing and Buffering for Delivery of Prerecorded Compressed Video", In Proceedings of *IS&T/SPIE Multimedia Computing and Networking* , Feb. 1995, San Jose, CA, pp. 234-242.

[4] Wu-chi Feng, F. Jahanian, S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video", *ACM/Springer-Verlag Multimedia Systems Journal,* September 1997.

[5] D.P. Heyman, T.V. Lakshman, "Source Models for VBR Broadcast-ideo Traffic," *IEEE/ACM Transactions on Networking*, Vol. 4, No. 1, pp. 40-48, Feb. 1996.

[6] Raj Jain, *The Art of Computer System Performance Analysis; Techniques for Experimental Design, Measurement, Simulation and Modeling*, John Wiley and Sons, 1991.

[7] Predrag Jelenkovic, A. Lazar, N. Semret, "The Effect of Multiple Time Scales and Subexponentiality in MPEG Video Streams on Queueing Behavior", *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 5, May 1997.

[8] Marwan Krunz, A. Makowski, "A Source Model for VBR Video Traffic Based on M/G/∞ Input Processes," In *Proceedings of INFOCOM '98*.

[9] A. Lazar, G. Pacifici, and D. Pendarakis, "Modeling Video Sources for Real-Time Scheduling," *ACM Multimedia Systems*, Vol. 1, No. 5, pp. 253-266, 1994.

[10]Didier LeGall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*, Vol. 31, No. 4, pp. 47-58, April 1991.

[11]B. Ryu, A. Elwalid, "The Importance of Long-Range Dependence of VBR Video Traffic in ATM Engineering: Myths and Realities," In *Proceeding of ACM SIGCOMM '96*, pp. 3-14, Aug. 1996.

[12]J.D. Salehi, Z.L. Zhang, J.F. Kurose, D. Towsley, "Optimal Buffering for the Delivery of Compressed Prerecorded Video", in *Proceedings of ACM SIGMETRICS*, pp. 222-231, May 1996.