MODELING FULL-LENGTH VBR VIDEO USING MARKOV-RENEWAL-MODULATED TES MODELS

Benjamin Melamed

Rutgers University
Faculty of Management
Department of MSIS
94 Rockafeller Rd.
Piscataway, NJ 08854

Dimitrios E. Pendarakis

IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Abstract

This paper describes a general methodology for constructing accurate source models of long sequences (full-length movies) of full-motion compressed VBR (variable bit rate) video, directly from empirical data sets of observed bit rate (frame size) records. The main idea is to first cluster scenes into classes and use TES (Transform-Expand-Sample) processes to carefully model their duration and bit rate processes, and then modulate these scene class models by a Markov-Renewal process that governs scene transitions. The resulting model is a Markov-Renewal-Modulated TES (MRMT) process. The fidelity criteria imposed are relatively stringent in that they call for the simultaneous capture of empirical histograms and autocorrelation functions, as well as the qualitative appearance of the empirical data, including burstiness. To illustrate the methodology, the paper provides a detailed description of the modeling procedure, using a JPEG-like coding trace of the "Star Wars" movie as a working example. The resulting MRMT model was validated by comparing its simulation-based statistics to their empirical counterparts, and by a performance study of the corresponding loss rates for various buffer sizes. The results support the efficacy of the modeling methodology and suggest that suites of realistic and compact source models can be developed for other types of movies under a variety of coding schemes. These may be used to drive realistic Monte Carlo simulations of broadband telecommunications networks.

Keywords and Phrases: Compressed Video, Coded Video, Markov-Renewal-Modulated TES Models, MRMT Processes, Simulation Modeling, TES Processes, Variable Bit Rate Video, VBR Video

1 Introduction

This paper is motivated by the need to create accurate source (traffic) models of video workload in terms of VBR (variable bit rate) compressed video. The term VBR refers to the fact that the bit rate (frame sizes) of compressed (coded) video is not constant, but rather a random process. It is expected that video traffic will constitute a significant portion of the load in future B-ISDN (Broadband Integrated Services Digital Networks), especially with ATM (Asynchronous Transfer Mode) transport, to be collectively referred to as *broadband networks*, for short. The size of broadband networks, combined with the complexity and diversity of their projected traffic mix (voice, video, file transfer, etc.), promise to render the task of network design and management a hard one. A careful performance analysis effort is required for accurate modeling of broadband networks, mainly via simulation experiments.

Effective broadband network simulations must rely critically on high-fidelity traffic modeling. More specifically, a fundamental aspect that must be faithfully captured by traffic models is traffic burstiness. This concept is easily grasped intuitively, but surprisingly difficult to characterize mathematically. Intuitively, burstiness is present in a traffic process, if arrival points appear to form visual clusters on the timeline. The mathematical underpinnings of burstiness are complex. Two main factors contributing to traffic burstiness are the marginal distribution (first-order statistics) and autocorrelation function (second-order statistics) of traffic interarrival times, particularly strong positive short-term autocorrelations. For additional discussion of traffic modeling and burstiness, see [7, 20].

The impact of autocorrelation in traffic processes on queueing performance measures such as mean queue length, mean waiting times and loss probabilities in finite buffers, can be very dramatic. A typical example is the simulation study in [28] which investigated the effect of introducing increasing autocorrelations into the Poisson process at an $M/M/1$ queue, without altering the (exponential) marginal interarrival time distribution. When offered to statistically identical exponential servers, the two traffic streams gave rise to mean waiting times differing by orders of magnitude, even in light traffic regimes! The same phenomenon has been observed in analytical queueing models, for instance [10, 15, 16, 25, 36, 40]. Note that the timeliness constraints of video imply that the main quality of service measures of interest are those involving loss measures.

The studies cited above support the view that modeling an autocorrelated traffic stream (interarrival times or bit rates) as a renewal process is fraught with modeling risk. Worse still, a renewal traffic model is bound to err on the optimistic side, if the modeled traffic has appreciable short-term positive autocorrelations. In fact, positive autocorrelations are prevalent in VBR video streams. The reason is quite straightforward: with the exception of scene changes, the nature of full-motion video pictures and their high generation rate (24 - 30 frames/second) tend to produce rather similar bitmaps in successive video frames. Consequently, a coded video sequence from a single scene possesses an inherent temporal dependence, which is statistically manifested by strong short-term autocorrelations in the random bit rate process; the latter are discernible to the human eye as "burstiness". Thus, one is led to the conclusion that autocorrelation modeling of compressed video is essential to high-fidelity modeling of broadband networks. The attendant modeling cost is added modeling complexity.

The bulk of previous work in video source modeling has been largely confined to short sequences of empirical records consisting of one or few scenes [32, 43, 44], or to conference video [11, 12, 37, 41]. The latter differs from full-motion video in two major ways: first it has relatively low bit rate, and second, scene changes are rare. The most common approaches employed in VBR video modeling are based on fluid models and autoregressive models. In the fluid approach, a video stream is modeled as having a fixed rate, modulated by some stochastic process; see, e.g., Elwalid and Mitra [5] which derives effective bandwidths for Markov fluid models. In the autoregressive approach, a video bit rate is modeled as a weighted sum of a finite number of previous bit rates. The DAR (Discrete Autoregressive) model was employed by Frater et al. [6] to model full-motion movies, including "Star Wars", and by Heyman et al. [12] to a variety of conference video records. A composite autoregressive scheme is used by Ramamurthy and Sengupta [37] to fit the observed autocorrelation function of teleconference video.

More recently, the increased availability of relatively long full-motion VBR video records rendered their modeling an active field of research. The "Star Wars" movie has been a popular modeling target, mainly due the fact that as an "action" movie, it contains a wide variety of scene types over a broad range of bit rates. Garrett [8] contains a detailed statistical analysis of the "Star Wars" VBR video empirical record under various codings, including the one used in our paper (*ibid.*, Chapter 4). Long-range dependence and heavy-tailed marginals in video records are discussed in [8, 1, 9]; the latter also investigates heavy-tailed marginals and concludes that statistical multiplexing can be achieved in the presence of long-range dependence. The "Star Wars" movie is used in [34] to argue that VBR video is preferable to CBR (constant bit rate) video. A description of MPEG-coded "Star Wars" statistics appeared in [35]. A comprehensive study of some eleven full-motion video sequences (including "Star Wars") and one teleconference video appeared in Heyman and Lakshman [13]. This study concluded that no single parsimonious family of models can currently capture the full range of video sequences; the differences between full-motion and teleconference video were noticed earlier in [42]. An abbreviated version of this paper appeared in [33].

Recently, a model of MPEG-1 video, based on GOP (group-of-picture) observations (sum of pictures in I-frame to I-frame cycles), has been proposed in [24]. The traffic model is a discrete-time variant of a semi-Markov process with iid GOP bit rates in each renewal interval, and is based on a mix of long video sequences (sportscasts, movies, music videos, newscasts, talk shows and cartoons). It aims to capture multiple time scales and subexponential characteristics (in the distribution tail and autocorrelation function tail). Three aspects of [24] are similar to our earlier work in [33].

First, scenes are segmented and clustered by average GOP size within scenes; second, scene distributions and autocorrelations are fitted to empirical data; and third, scene durations are modeled as a renewal process. Reisslein and Ross [39] focus on a VBR traffic trace of MPEG-1 "Star Wars", and take random starting points in order to convert the trace into multiple traces. They multiplex those streams in a bufferless transmission model that investigates admission control policies. A short sequence (3.5 minutes) of "Star Wars" under MPEG and JPEG compressions is modeled in [2]. The model is fitted to the empirical rate histogram and power spectrum, and studies admission control performance of two cell-multiplexing scheduling algorithms.

One potential drawback of traditional autoregressive and Markovian models is that they lack a systematic way of simultaneously fitting the empirical marginal distribution and autocorrelation function. In this paper we employ a versatile class of transformed autoregressive processes, called *TES (Transform-Expand-Sample)* processes (see Section 6.1), which are designed for such simultaneous modeling and have been used to model VBR video at the frame level and the group-of-block level [32, 30], as well as H.261 coding [27] and MPEG-1 coding [38]. A brief review of traffic processes, video included, may be found in [7], and a more extensive one in [21].

This paper is a detailed version of [33]. It presents a general methodology for modeling the bit rate process of *full-length movies*, i.e., of long video sequences consisting of a mix of scene types, in the sense that constituent scenes can be categorized into distinct classes having markedly different statistical signatures. The paper's main contribution is the melding of TES models with the Markov modulation scheme: TES processes are used to model carefully the bit rate process of a small number of scene classes, while a Markov-Renewal process is used to model scene transitions.

The modeling methodology consists of five steps.

1. The entire movie is segmented into scenes.
2. The set of all scenes is partitioned into a small number of classes by a clustering algorithm. For the "Star Wars" movie, the algorithm yielded just four classes.
3. A class tag is assigned to each scene. The resulting random sequence of scene class tags is used to construct a simple Markov chain whose states correspond to the classes obtained from the clustering algorithm.
4. The bit rate process of each scene class is modeled as a TES process — one per class.
5. The sequence of scene durations for each class is modeled as a TES process. For the "Star Wars" movie, each duration process was modeled as a renewal process — a special case of TES processes.

It is convenient to describe the model operation from a simulation perspective. Initially, a class tag is sampled from the stationary distribution of the Markov chain, and a scene duration is sampled from the associated duration process. Next, the

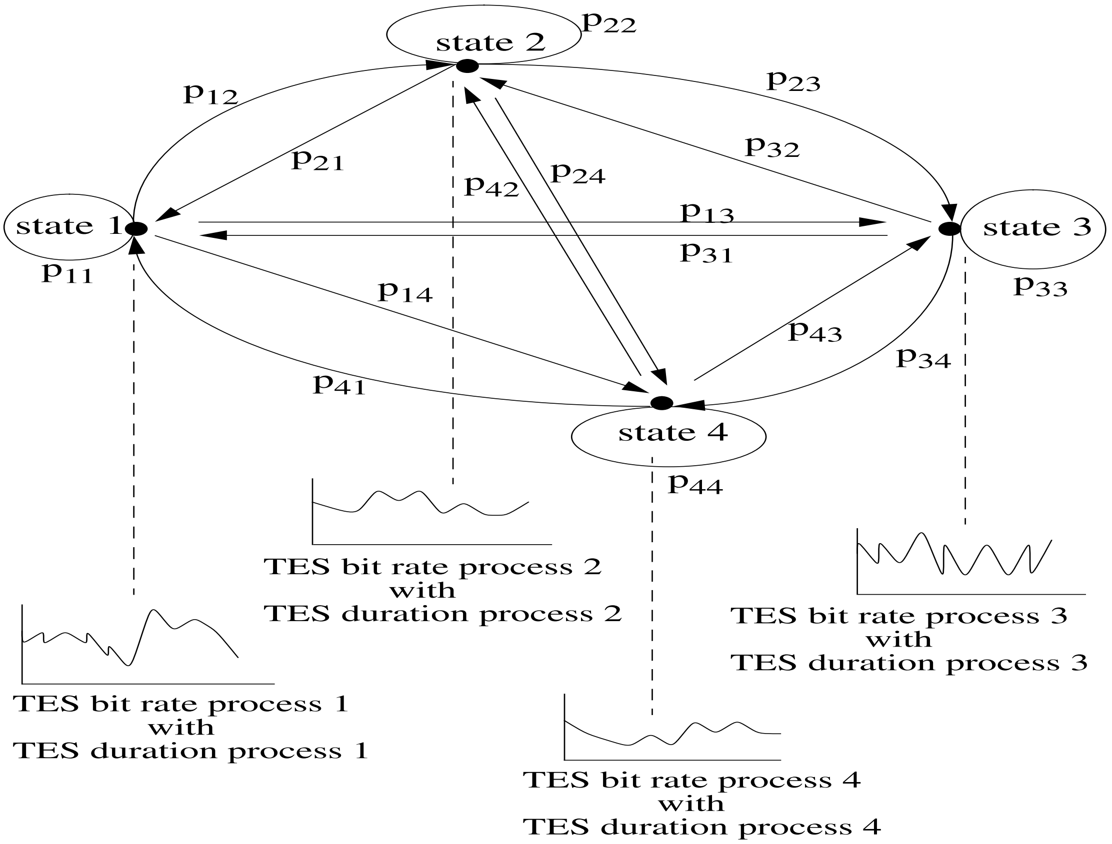


Figure 1: The Markov-Modulated TES model with TES durations (MMTT).

corresponding TES process of bit rates is run for that scene duration. When this duration expires, the next class tag is sampled from the transition matrix of the modulating Markov chain, given the previous class tag. This determines the probability law for sampling the next scene duration and next bit rate process, and so on. The general structure of this model is depicted in Figure 1. The states of the Markov chain represent class types, and its stochastic evolution modulates scene classes. More precisely, the prevailing Markovian state determines the probability law of the current scene duration variate and the current scene bit rate process for that duration; scene transitions correspond to state transitions in the modulating Markov chain. We shall refer to this class of models as *Markov-Modulated TES Traffic with TES Duration* process (MMTT, for short). Thus, the MMTT model generalizes the well-known Markov-Modulated Poisson Process (MMPP), which had been successfully applied to model voice traffic in telephony [10]. The "Star Wars" model to be described in the sequel is a natural special case, to be referred to as a *Markov-Renewal-Modulated TES (MRMT)* process, by virtue of the renewal property of the durations and their dependence only on the current state (class tag).

The model proposed here relies on the TES modeling methodology (to be explained later) to accurately capture both first-order and second-order statistics of the empirical bit rate record; more specifically, the model approximates well the marginal distribution and the autocorrelation function (equivalently, spectral density) of the bit rate process, *simultaneously*, and model sample paths bear a strong *"resemblance"* to the empirical record. Note, though, that sample path "resemblance" is a subjective judgment which cannot be precisely quantified. While its exact meaning is best left at the intuitive level, we point out that such judgments are routinely made by analysts in an informal manner.

The rest of this paper is organized as follows. Section 2 describes the compression algorithm which gave rise to the empirical bit rate record used. Section 3 explains the algorithm used to segment the empirical record into scenes, and Section 4 describes the clustering procedure used to classify individual scenes into scene classes. Section 5 describes the construction of the modulating Markov chain, while Section 6 details the modeling of the bit rate process of individual scene classes via TES processes, and the corresponding scene durations via renewal processes. Finally, Section 7 reports the results of the modeling study and a performance study of loss rates.

2 The Empirical Coded Video Data of "Star Wars"

The empirical coded VBR video data, used in this study, [1] was the bit rate record obtained from compressing the "Star Wars" movie via a variant of the JPEG coding scheme [9]. This movie is a good representative of full-motion video, since it contains a diverse collection of video sequences whose visual complexity and motion content range from low to high and very high. The data record consisted of some 171,000 frame sizes (in bytes per frame) — the equivalent of about two hours of video elapsed time at the rate of 24 frames a second.

The coding scheme employed is a simplified variant of JPEG. Following is a description of this scheme, taken from Garrett [8]. The simplified variant coded only the luminance component, resulting in monochrome video. The digitized frame size is 480×504 pixels; the last 8 pixels on each 512-pixel line form the horizontal retrace interval, and those are dropped before coding. The frame is partitioned into blocks of 8×8 pixels, on which a discrete cosine transform (DCT) is performed. The DCT coefficients are uniformly quantized into 8 bits and then entropy coded. The Huffman code table includes 64 events to represent runs of from 1 to 64 consecutive zeros, because a block contains 64 coefficients. The remaining 255 events are for the non-zero coefficient values. The (fixed) codewords are generated from the relative frequencies of each event as measured in a representative 2-minute segment of the movie (specifically frames 2851 through 5700). No differential or motion-compensated coding were included. When compared to coding schemes with tables derived from different 2-minute segments, the average bit rate varied by less than 5% — a good approximation of the original bit rate. A detailed description of the empirical bit rate record appears in Garrett [8].

---

[1] The record was provided to us courtesy of Mark Garrett, Bell Communications Research (Bellcore).

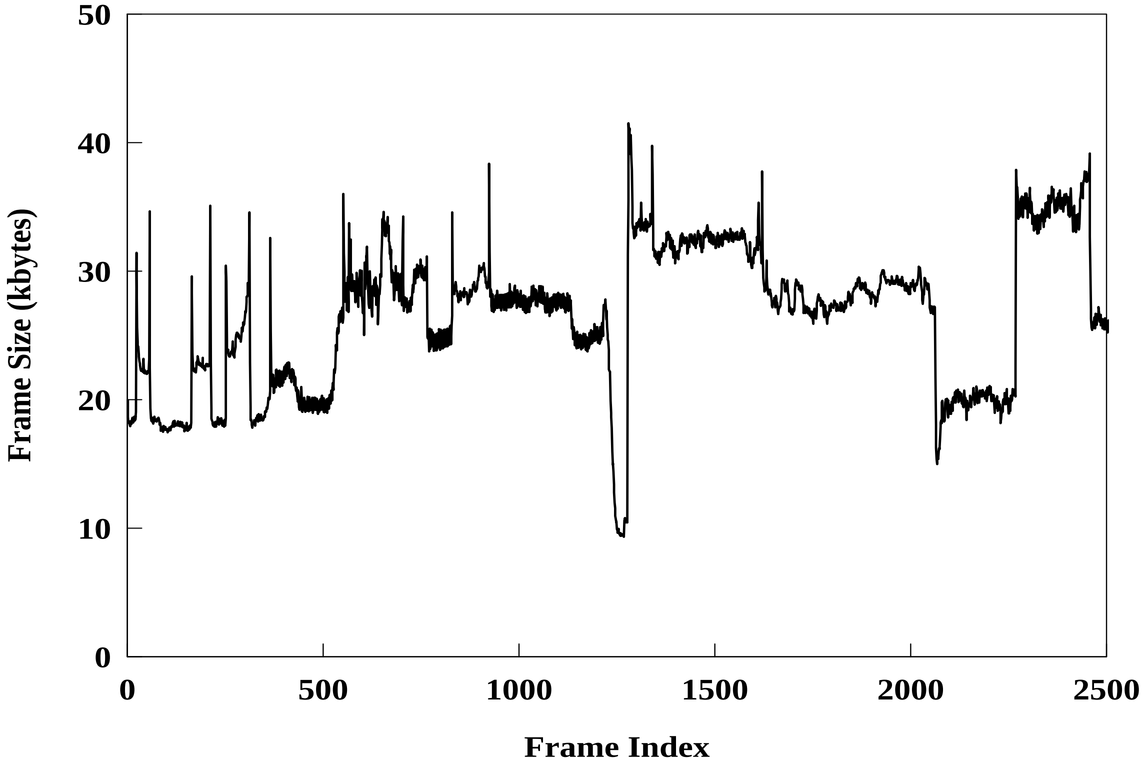


Figure 2: A low activity segment from the “Star Wars” bit rate record.

3 Segmenting the Bit Rate Stream Into Scenes

Figure 2 depicts a representative portion of the empirical bit rate record. The VBR sequence of frame sizes, $\{X_n\}$, exhibits high short-term autocorrelations, as suggested by the fact that short runs of successive frame sizes tend to lie close to each other. This heuristic observation is borne out by calculating estimates of short-term autocorrelations from the empirical record.

The appearance of the data plotted in Figure 2 suggests a structure of punctuated stationarity. In other words, the record appears to be composed of stationary segments, with random transitions among segments. The boundaries of these transitions would be visually characterized by abrupt large magnitude changes in the local average of frame sizes. Intuitively, one would expect to relate visually-discernible segments in the empirical record to a physical phenomenon in the underlying movie. Specifically, we wish to establish that these segments do correspond to distinct *scenes* of the movie, i.e., portions of the movie without significant shifts of view; such shifts result from abrupt camera zooming, panning, or cutting to a new visual context. By observing the actual movie frames and the corresponding bit rate sequence in lockstep, it was possible to ascertain that scene changes do indeed coincide with abrupt sustained changes in bit rate magnitude.

In order to detect scene boundaries in the empirical bit rate record, a methodology similar to that introduced in [26] was followed: A scene change is declared in the bit rate stream when the absolute value of the difference between successive bit rates exceeds a certain threshold in a sustained manner. To formalize this notion, let J_{min} denote a threshold. The random sequence of bit rate jumps, $\{J_n\}$, which exceed the J_{min} threshold, is given by the characteristic function

$$J_n \ = \ 1\{|X_n - X_{n-1}| > J_{min}\}. \tag{1}$$

A careful look at the empirical record reveals, however, that not all jumps exceeding J_{min} actually correspond to scene changes. Rather, it was observed that the first frame following a scene change usually had a bit rate, unrelated to that of the following or preceding frame. Thus, in order to correctly identify scene changes, it is necessary to exclude jumps from the first bit rate to the second one as scene boundary candidates. The solution is to look for sustained bit rate changes by defining a *minimum scene length*. The indicator function sequence, $\{S_n\}$, of frame indices corresponding to scene changes is then given by

$$S_n \ = \ 1\{J_n = 1, \ J_{n-1} \neq 1, \ J_{n-2} \neq 1, \ldots, J_{n-L_{min}} \neq 1\}, \tag{2}$$

where L_{min} is the minimal scene length in frames. In the "Star Wars" case, the values of $\{S_n\}$ were obtained by setting $J_{min} = 5000$ bytes and $L_{min} = 2$ frames. With these parameters, the sequence $\{S_n\}$ captured well scene changes in the empirical record.

4 Classifying Scenes into Classes

Having segmented the empirical bit rate sequence into individual scenes, it clearly is neither practical nor parsimonious to model each individual scene separately. The modeling complexity will become manageable, though, if the set of all scenes were partitioned into a small number of equivalence classes. Operationally, classes would be selected so as to group together scenes with common statistical characteristics; mathematically, all scenes in a given class would be assumed to be realizations from a common stationary probability law. Each equivalence class of scenes would then be modeled by one bit rate process and one duration process, designed to capture the "common" statistical signature of its scene members (marginal distribution, autocorrelation function, and sample path "resemblance"). Finally, the obtained equivalence classes would be identified with the states of some Markov chain (to be constructed in Section 5), and their labels (e.g., natural numbers) would be used to tag scene classes.

Accordingly, assume that the empirical record consists of scene realizations which can be classified into a small number of stationary probability laws (classes); assume further that scene bit rates and durations, as well as scene transitions, are governed by a simple Markov chain whose states coincide with partition class tags, as in Figure 1. The partitioning and classification problem can be broken down into the following stages:

1. Identify a reasonable set of partitioning criteria.
2. Select a partitioning methodology yielding a small number of equivalence classes.
3. Implement a procedure for classifying scenes into these classes, based on the aforementioned partitioning criteria and methodology.

In Stage 1, it was decided to use either scene bit rate mean or scene bit rate variance (or both) as partitioning criteria. As it happened, scene bit rate mean alone was found to be a good criterion for characterizing scene classes (see below). Intuitively, this criterion classifies scenes based on their "activity level".

In Stage 2, *partitional clustering* was selected as a partitioning methodology [22]; henceforth, the terms 'classes' and 'clusters' will be used interchangeably, depending on the context. The general abstract clustering problem is formulated as follows. Given n elements in a d-dimensional metric space, find a partition of the elements into K clusters, such that elements in a cluster are "closer" to each other than to elements in all other clusters; the value of K may or may not be prescribed. A clustering metric formalizing the notion of "proximity" has to be selected. To fix the ideas, we shall henceforth deal solely with the most common metric used in practice (and the one used in this study), namely, the Euclidean distance

$$\eta(\mathbf{v}, \mathbf{w}) \;=\; \sqrt{\sum_{i=1}^{d} (v_i - w_i)^2},$$

for vectors $\mathbf{v} = (v_1, \ldots, v_d)$ and $\mathbf{w} = (w_1, \ldots, w_d)$ in d-dimensional Euclidean space. The clustering methodology is an iterative heuristic, proceeding in a trial-and-error fashion. Assume that the n elements from the d-dimensional Euclidean space have already been partitioned into K clusters, $\{C_1, C_2, \ldots, C_K\}$ and enumerated within clusters, such that cluster $C_k = (\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{n_k}^{(k)})$ is of size n_k ($\sum_{k=1}^{K} n_k = n$) and $\mathbf{x}_j^{(k)}$ is the j-th element in cluster C_k. The mean vector, or center, of cluster C_k is defined as the *cluster centroid*,

$$\mathbf{m}^{(k)} \;=\; \frac{1}{n_k} \sum_{j=1}^{n_k} \mathbf{x}_j^{(k)}, \tag{3}$$

where all vector operations are performed component-wise. The square-error, e_k^2, for cluster C_k is the *within-cluster variation*,

$$e_k^2 \;=\; \sum_{j=1}^{n_k} \eta^2(\mathbf{x}_j^{(k)}, \mathbf{m}^{(k)}), \tag{4}$$

while the square error for the entire K-cluster partition is the *sum of within-cluster variations*,

$$E_K^2 \;=\; \sum_{k=1}^{K} e_k^2. \tag{5}$$

For any given K, the clustering heuristic searches the partition space of K clusters, attempting to find one which minimizes E_K^2.

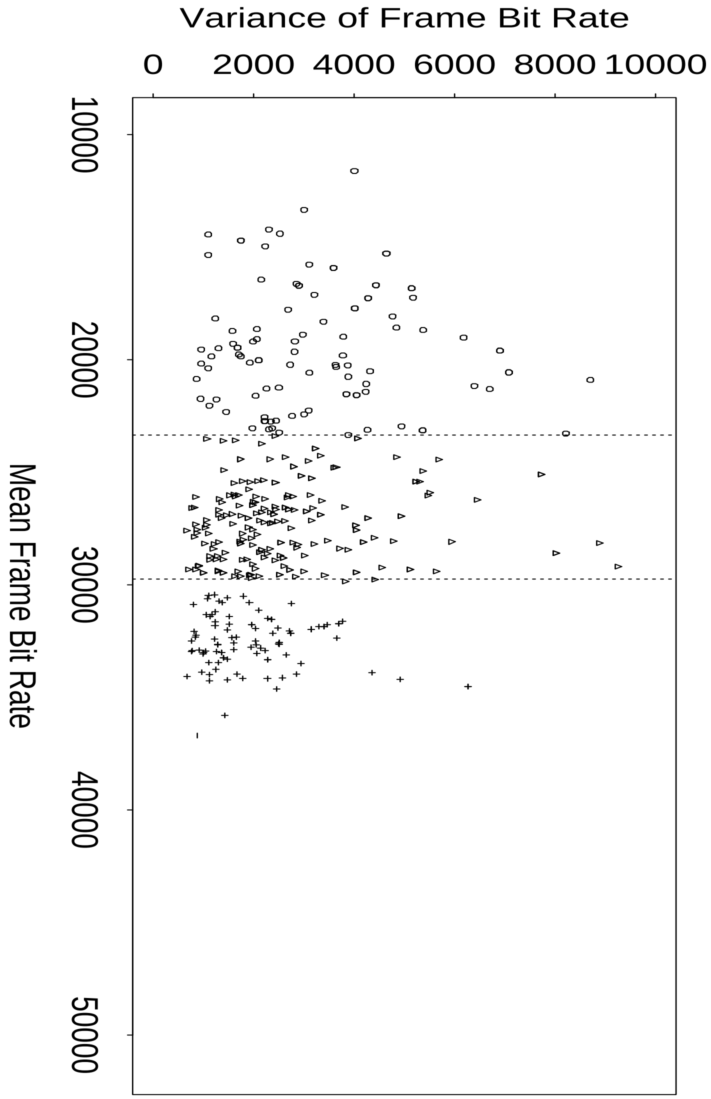
In Stage 3, a heuristic *clustering procedure* was implemented and run, to search for a reasonable number of clusters, K, as well as a suitable partition. The procedure is outlined below.

The Clustering Procedure

1. Select an initial partition with K clusters. The standard initial partition consists of singleton clusters.
2. Generate a new partition by assigning each element (in our case, scene) to the cluster with the nearest centroid (3).
3. Compute the new cluster centroids from (3), and the associated square-error (5), using (4).
4. Repeat steps 2 and 3 until the square-error (5) converges.
5. Adjust the number of clusters by merging and splitting existing clusters or by removing small or outlier clusters.

The last step above requires the specification of two auxiliary parameters: the *minimum distance between cluster centroids*, η_{min}, and the *maximum variation within a cluster*, E^2_{max}. These two parameters are used as follows. If the distance between two cluster centroids drops below η_{min}, these clusters are merged into one; in a similar vein, if the variation within a cluster exceeds E^2_{max}, that cluster is split into two clusters. By varying the two auxiliary parameters, one can control the total number of clusters obtained from the procedure above. It should be clear, however, that convergence of the procedure is not guaranteed for all possible combinations of the auxiliary parameters, due in part to their dependence: Increasing η_{min} effectively forces more clusters to be merged, and consequently increases the variation within individual clusters. Furthermore, resultant partitions may well correspond to local, rather than global, minima of the squared error. In practice, however, a large number of iterations often attains a global minimum.

The implementation of the clustering methodology above attempted initially to classify "Star Wars" scenes by the (mean, variance) pairs of scene bit rates. In other words, the initial clustering criteria gave rise to a clustering problem in the Euclidean plane. Figure 3 depicts partial results obtained via the Clustering Procedure; for typographical clarity, this figure displays the partition computed just for the first 500 scenes (out of a total of 2198 scenes), as plotting the entire set of scenes results in excessive clutter. The auxiliary parameters were set to $\eta_{min} = 4600$ and $E^2_{max} = 10,000$, both in bytes-per-frame units. A partition into four clusters is depicted in Figure 3, each marked by a distinct character. The main point made by this figure is the significance of the mean scene bit rate as a clustering criterion, and the comparable insignificance of the corresponding variance. To see that, note that vertical lines have been drawn in Figure 3 to demarcate approximate cluster boundaries by mean bit


Variance of Frame Bit Rate
0
2000
4000
6000
8000
10000
Mean Frame Bit Rate
10000
20000
30000
40000
50000