MPEG4 Traffic Modeling Using The Transform Expand Sample Methodology

Ashraf Matrawy Ioannis Lambadaris Changcheng Huang

Broadband Networks Laboratory
Department of Systems and Computer Engineering
Carleton University
{amatrawy, ioannis, huang}@sce.carleton.ca
Phone: +1-613-520-2600 ext. 2913
Fax: +1-613-520-5727

Abstract

The transfer of digital video will be a crucial component of the design of future home networking applications. This transfer was made feasible by the advancement of digital video encoding techniques that reduced the bandwidth required for this transfer to a practical level. MPEG4 is an encoding technique that is suitable for home networking applications with its low bit rate. It also has the advantage that allows viewers to interact with encoded objects. In this paper, we present our work that enables the study of MPEG4 properties and performance on the Internet using simulation. We propose a traffic generator that is able to generate traffic that has almost the same first and second order statistics as an original trace of MPEG4 frames that is generated using an MPEG4 encoder. We model and generate this traffic based on the Transform Expand Sample (TES) methodology using *TEStool*. We present the model and show the performance of the generator in terms of good matching of the characteristics of the modeled trace.

1. Introduction

The Internet's impact on daily life is increasing dramatically. Relying on the Internet as a source of information and a means of communication is unmatched in history. This is the reason why some researchers and communication experts believe that sooner or later, all the devices that are parts or our daily life, like home appliances, will be connected to the Internet. Connecting those devices and appliances to the Internet will require the transfer of video or at least will include video transfer as an enhancement. This is one of the motivations behind establishing the ISO Moving-Picture-Experts-Group (MPEG) family of standards for digital video encoding. This family of standards includes MPEG1, MPEG2, and MPEG4. MPEG7 is under development.

MPEG4 is the one that is most suitable for the Internet [1]. It is targeted for low bit rates. It allows real images to co-exist with computer-generated counterparts and also allows their separation and their receiving different treatment due to interaction with the user. The main feature of importance to the network is MPEG4's capability of real-time adaptive encoding. This enhances network utilization and enables MPEG4 senders to be more responsive to changes in network conditions. It generates video in three different frame types (I, P, and B) that serve to encode different portions of the video signal in different levels of quality.

In this work, we present our traffic generator that can be used for studying MPEG4 behavior and performance through simulation. This is relevant to giving a more realistic sense to simulations of multimedia transfer on the Internet. The traffic we generate closely matches the statistical characteristics (in terms of marginal distribution and auto-correlation function) of an original real trace of video frames that were generated using an MPEG4 encoder. We developed a model using the Transform Expand Sample (TES) [2], which is a versatile methodology for modeling any set of given observations in a time series. We modeled I frames, P frames, and B frames using three different TES models. We then used interleaving to generate the original sequence of frames for MPEG4. We implemented this generator in software and integrated it into the network simulator *ns-2* [3].

The rest of this paper is organized as follows: Section 2 gives an overview of the MPEG4 encoding technique for digital video and how it fits into a networked appliances environment. In Section 3, we present the Transform Expand Sample (TES) models and their applications. Section 4 gives the details of how we modeled MPEG4 using TES and demonstrates how closely the generated traffic matches the real traffic. Section 5 concludes the paper and presents our future research in this area.

2. MPEG4 video encoding

In [4], where requirements for networked appliances were presented, the specification of Home Audio and Video Interoperability was a main requirement. Also, support for media streaming was one of the communication mode requirements. MPEG4 has a lot of features that may fulfil these requirements [1]:

- It is targeted to provide high video quality at relatively low bit rates. This makes it suitable for home devices that does not have high speed access to the network
- It allows interactivity of the users at the decoder side
- Scalable content, which means encoding once and playing back at different speeds
- Real-time adaptive encoding that results in changing the sending rate in reaction to changing network conditions
- Scalable encoding that differentiates between important objects and less important objects. This results in streams that are suitable for transmission over networks that support priorities

MPEG4 encoders generate three types of frames: I frames, P frames and B frames. They can be described as follows:

- I frames (intra-coded frames) contain the information that results from encoding a still image, i.e., with no reference to any other image. They are points of reference and random access in the video stream. They can be decoded without the need for any other frames. The compression rate for these frames is the lowest of all frame types.
- P frames (predictively coded frames) require information from previous I frames and/or P frames for encoding and decoding. By exploiting temporal redundancies, P frames achieve higher compression rates than that for I frames.
- B frames (bidirectionally predictively coded frames) require information from the previous and following I frames and/or P frames for encoding and decoding. They are predicted from both previous and following frames. They have the highest compression ration among all frame types.

Reference frames must be transmitted first. Therefore the transmission order and display order may differ, as described in Fig. 1. At the beginning, there is always an I frame. The first I frame and the first P frame serve as a reference for the first two B frames. At the first time, the first I frame is the reference for the first P frame. The second I frame serve as a reference for the B frames that are displayed before it, that's why it has to be sent first.

Transmission order: I P B B P B B P B B I B B

Display order: I B B P B B P B B P B B I

Fig. 1 Frame transmission and display orders

3. Transform Expand Sample (TES) models

TES is a versatile methodology for generating data that closely match (in terms of its marginal distribution and auto-correlation function) any set of given observation of a time series. The derivation of TES models is performed in two phases [2]. First, a correlated sequence, with uniform marginal in [0,1], (also called *background* TES process) is formed as follows:

$$U_n^+ = \begin{cases} U_0 & n = 0 \\ \langle U_{n-1}^+ + V_n \rangle & n > 0 \end{cases}$$

or

$$U_n^- = \begin{cases} U_n^+ & n \text{ even} \\ \langle 1 - U_n^+ \rangle & n \text{ odd} \end{cases}$$

Where U_0 is a random variable uniformly distributed in [0,1]. $\{V_n\}$ is a sequence of iid random variables independent of U_0, called the *innovation sequence* (with probability density f_v). The operator < > denotes the modulo-1 operation which for every real x is defined by:

$$\langle x \rangle = x - \max\{\text{integer } (n) : n \leq x\}$$

Both U_n^+ and U_n^- can generate lag-1 autocorrelations in the range [0,1) and [-1,0) respectively. Regardless if the probability law of the innovation density f_v, it can be shown [5] that the TES background sequence forms a stationary Markovian process with a uniform in [0,1) marginal distribution. A smoothing operation called the *stitching transform* is used to get rid of the inherent discontinuity of the < > operator [2]. So in short, in the first phase of the TES model development, the objective is to generate a time series of correlated random variables with uniform marginals [0,1). The amount of correlation depends on the structure of the density function f_v of the innovation sequence $\{V_n\}$, which in general may be arbitrary.

In the second phase, synthetic sample data, called the *foreground sequence*, which resembles the real sample, may be derived from the *background sequence* using the inversion technique in [5]. This inversion technique allows the transformation of any uniform random variable to one with arbitrary distribution. More specifically, it is assumed that a histogram *H(.)* of the empirical video sequence has been built consisting of J cells, where the j^{th} cell is positioned on the interval $[l_j, r_j)$ and is characterized by a probability p_j. Such a histogram can be inverted as follows:

$$H^{-1}(x) = \sum_{j=1}^{J} I_{[c_{j-1}, c_j]}(x) \left[l_j + (x - c_{j-1}) \frac{w}{p_j} \right]$$

Where I_A is the indicator function of the set A, $w_j = r_j - l_j$ is the width of cell j, and $c_j = \sum_{i=1}^{j} p_i$, $1 \le j \le J$, ($c_0 = 0$, $c_J = 1$). The random variate $\{H^{-1}(x)\}_{n=1}^{\infty}$ has a marginal distribution *H(.)* and autocorrelation that depends upon the innovation density f_V.

The modeling process is carried out in two steps: the first step essentially captures the autocorrelation of the input time series using innovation f_V and applying a smoothing technique using the stitching transform [5]. This would result in the generation of the *background sequence*. The second step is to invert the *background sequence* using the empirical histogram, which always guarantees the matching of the marginals. The task of finding a suitable f_V that approximates the empirical autocorrelation is carried out via a heuristic search using *TEStool* [5]. The matching of the histogram is guaranteed, while the matching of the autocorrelation is carried out in an interactive manner with visual feedback for the best f_V.

4. Modeling MPEG4 using TES

In this section, we present our model for MPEG4 traffic generation. We used a sequence of MPEG4 frames that was generated by encoding a news broadcast from the BBC at a rate of 30 frames/sec. Here are the steps we followed to generate the traffic:

- We used a trace of frame sizes (in bits) of an MPEG4 encoded video. Fig. 2 shows the characteristics of the trace as generated while developing the model using the *TEStool*. Window *w1* of the figure shows the sample path of the frame sizes, in *w2* the autocorrelation function is shown, while *w3* shows the histogram of the distribution of the frame sizes.
- We separated this trace into three traces, one for each frame type. This facilitates the generation of a trace that matches the original trace. Trying to do that with the original trace without separation proved to be very difficult. This is due to the periodic nature of the autocorrelation function of the original trace. Having done that we model each of the traces using a separate TES model using the methodology described in Section 3.

- Fig. 3, Fig. 4, and Fig. 5 show the models of the I frames, P frames, and B frames, respectively. In each figure, *w4* shows the innovation density f_v that was descried in Section 3. In all the figures, the generated data is shown in blue (or darker line) while the original is in green. From the figures, we notice that the histogram always matches the original data while there is a little difference in both the autocorrelation function and the sample path.
- To generate the MPEG4 trace, we interleave generating samples from the three developed models to match the sequence that is shown in Fig. 1. The final trace that is generated by interleaving can be compared to the original trace as shown in Fig. 6.

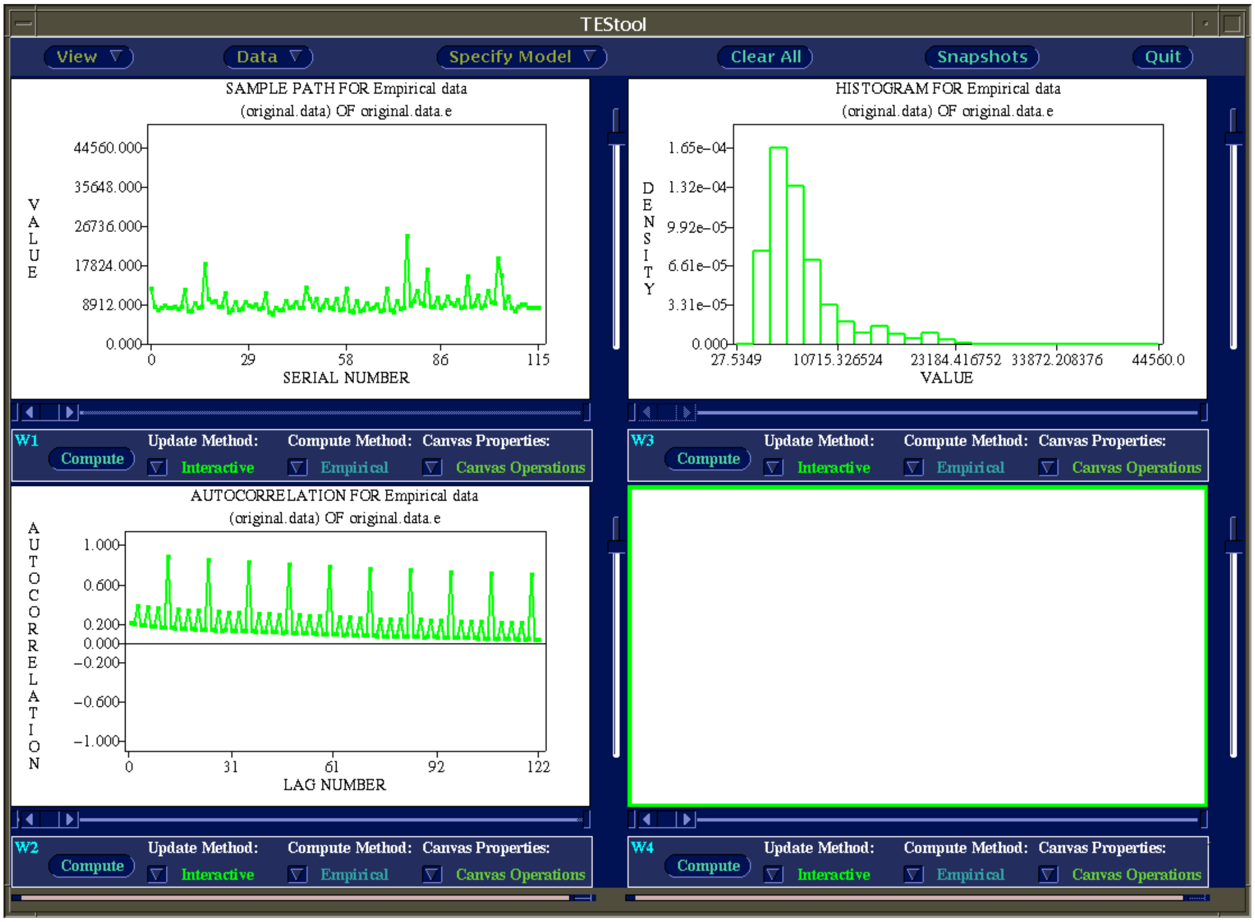
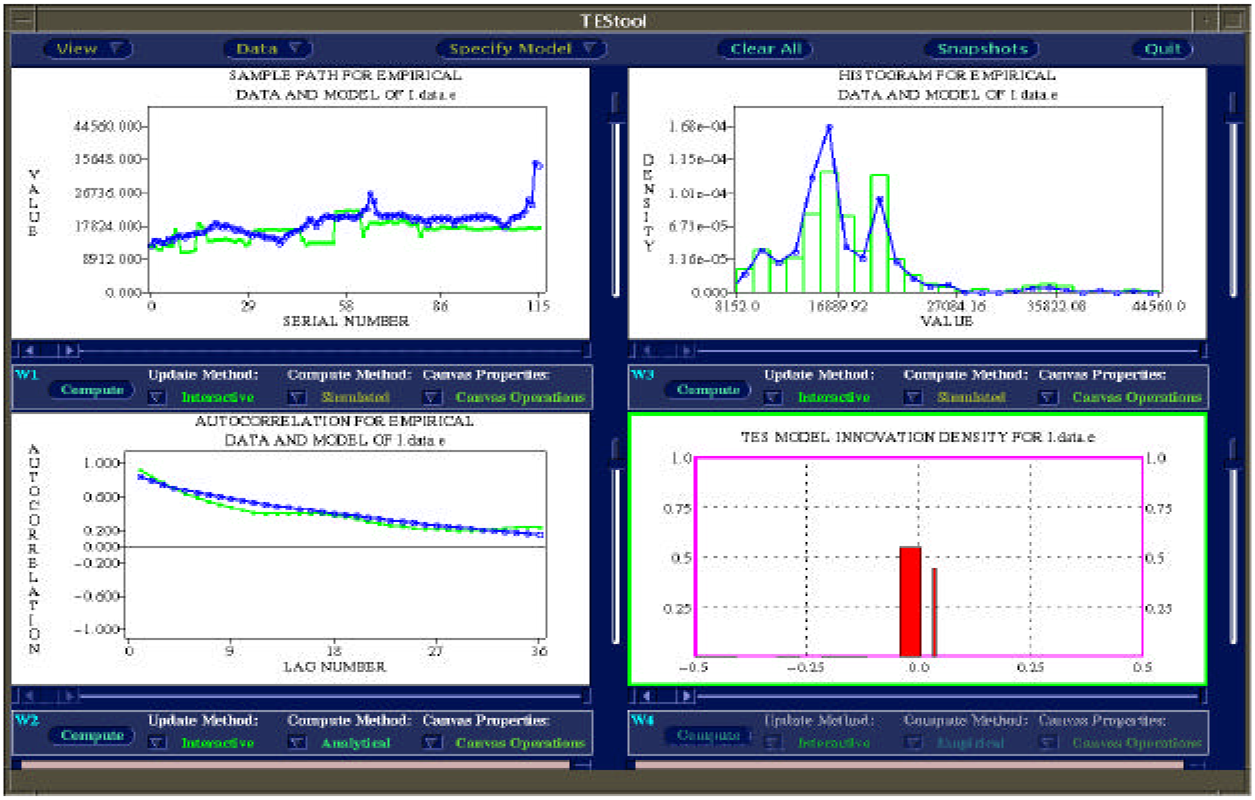


Figure 2: The original trace

- We implemented this generator in software. First, we use *TEStool* to generate the histograms and the innovation density for the three frame types. Then we provide this as an input to the software we wrote in C++ that uses each of the three models to generate a frame size of each type in the order mentioned in Fig. 1. We integrated this code into the *ns-2* simulator to generate traffic for simulation of IP networks.



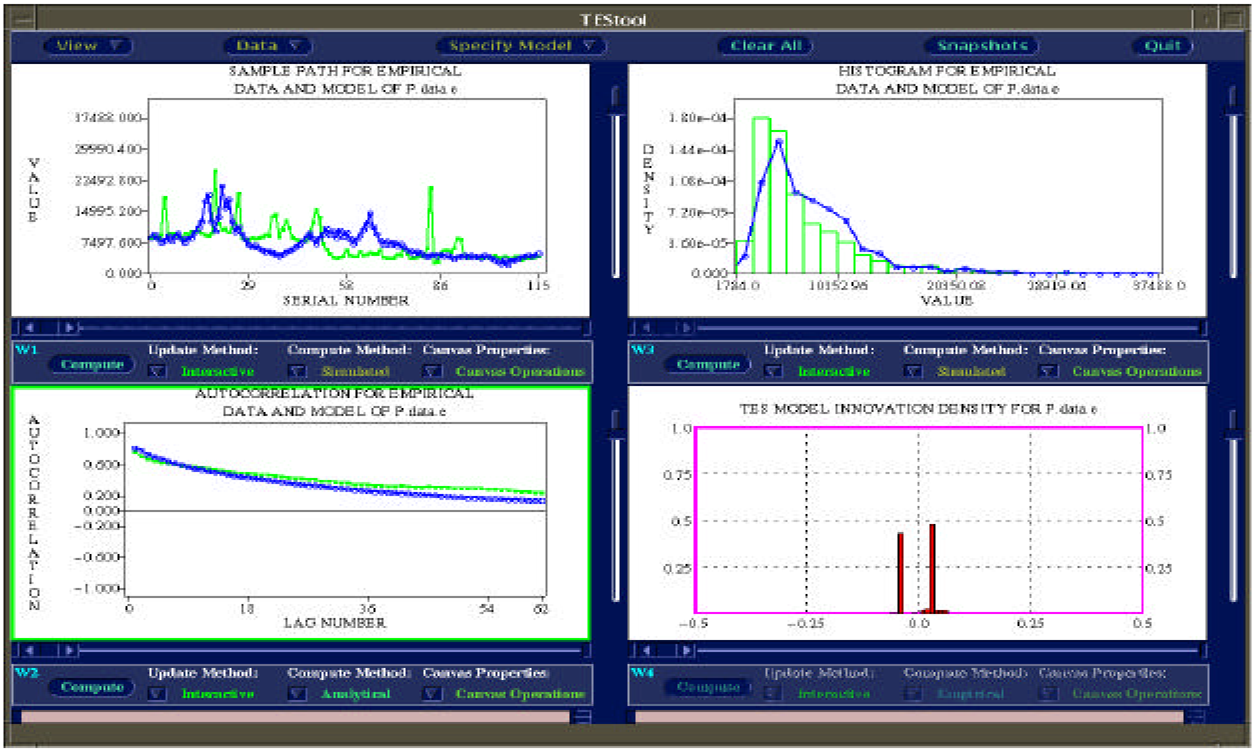
W1 — Update Method: Interactive; Compute Method: Simulated; Canvas Properties: Canvas Operations

W3 — Update Method: Interactive; Compute Method: Simulated; Canvas Properties: Canvas Operations

W2 — Update Method: Interactive; Compute Method: Analytical; Canvas Properties: Canvas Operations

W4 — Update Method: Interactive; Compute Method: Empirical; Canvas Properties: Canvas Operations

Figure 3: The I frames model



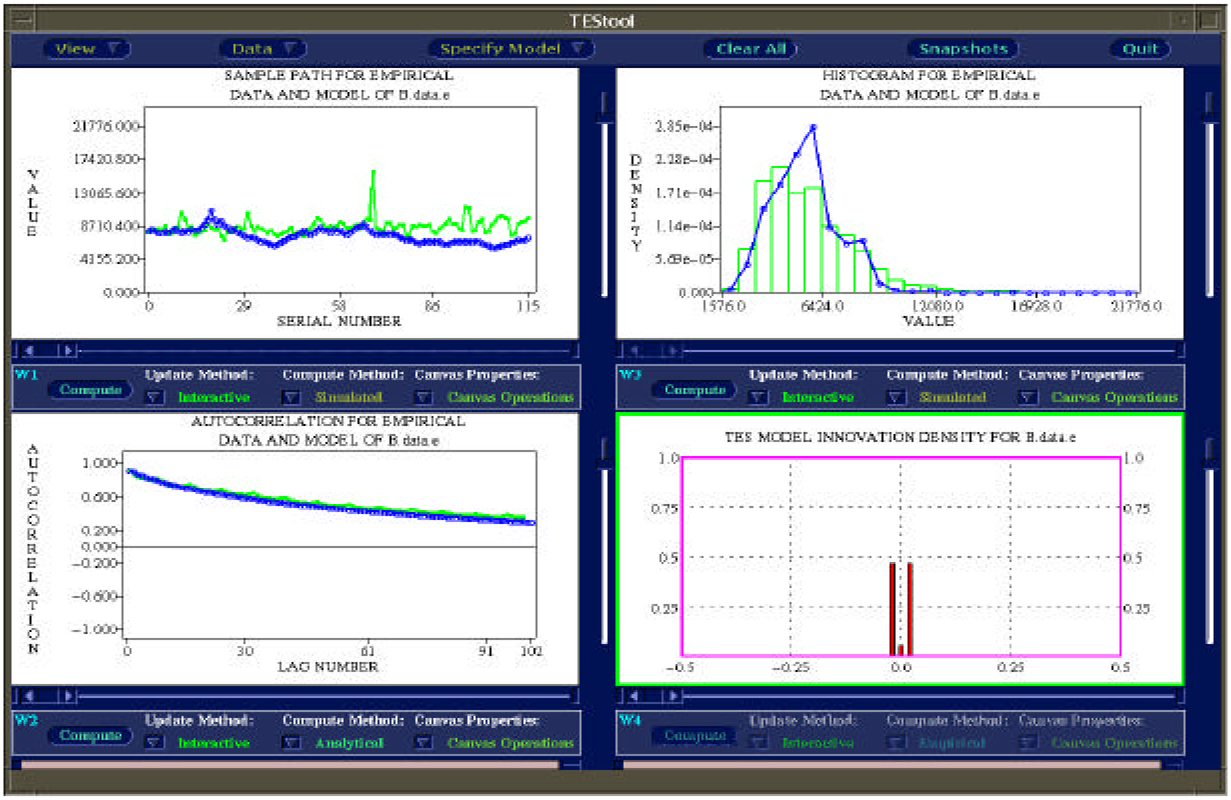
W1 — Update Method: Interactive; Compute Method: Simulated; Canvas Properties: Canvas Operations

W3 — Update Method: Interactive; Compute Method: Simulated; Canvas Properties: Canvas Operations

W2 — Update Method: Interactive; Compute Method: Analytical; Canvas Properties: Canvas Operations

W4 — Update Method: Interactive; Compute Method: Empirical; Canvas Properties: Canvas Operations

Figure 4: The P frames model



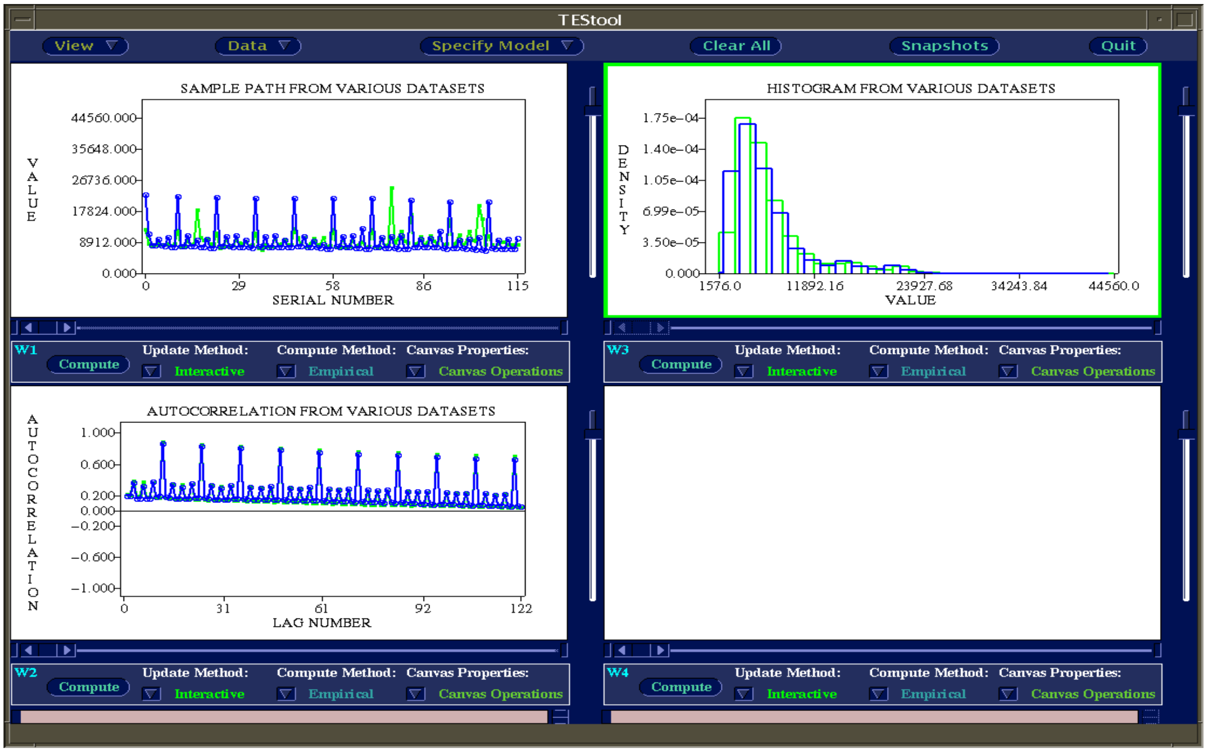
W1 — Update Method: Interactive; Compute Method: Simulated; Canvas Properties: Canvas Operations

W3 — Update Method: Interactive; Compute Method: Simulated; Canvas Properties: Canvas Operations

W2 — Update Method: Interactive; Compute Method: Analytical; Canvas Properties: Canvas Operations

W4 — Update Method: Interactive; Compute Method: Empirical; Canvas Properties: Canvas Operations

Figure 5: The B frames model



W1 — Update Method: Interactive; Compute Method: Empirical; Canvas Properties: Canvas Operations

W3 — Update Method: Interactive; Compute Method: Empirical; Canvas Properties: Canvas Operations

W2 — Update Method: Interactive; Compute Method: Empirical; Canvas Properties: Canvas Operations

W4 — Update Method: Interactive; Compute Method: Empirical; Canvas Properties: Canvas Operations

Figure 6: Comparison between the original and the generated traces

5. Conclusion and future work

We have presented our work on generating video traffic that matches the statistical properties (first and second order) of an original trace of MPEG4 frames. We used the Transform Expand Sample (TES) methodology to model the MPEG4 trace. We have successfully modeled the I, P, and B frame sequence and we showed how the generated traffic matches the original trace. This work is relevant to the simulation MPEG4 performance in networked appliances applications as well as any other video application that runs on IP networks. For future work, we will be developing models that may model more properties of the video trace such as generating different priority packets and capturing self-similarity of the video traffic.

References

1. R. Koenen, "MPEG-4 Overview", *IEEE SPECTRUM*, Vol. 36, No. 2, February 1999.
2. B. Melamed, "An Overview if TES Processes and Modeling Methodology", *In Performance Evaluation of Computer and Communication Systems*, 1993.
3. http://www.isi.edu/nsnam/ns/
4. S. Tsang, S. Moyer, D. Marples, H. Schulzrinne, and A. Roychowdhury, "Requirements for networked Appliances: Wide-Area Access, Control, and Interworking, *Internet Draft, draft-tsang-appliances-reqs-01.txt*.
5. B. Melamed et al., "TES-Based Modeling for Performance Evaluation of Integrated Networks", *In Proc. of INFOCOM*, 1992.