A QoS evaluation of video traffic models for H.264 AVC video

Savera Tanwir and Harry Perros  
Department of Computer Science  
North Carolina State University  
Raleigh, North Carolina  
Email: stanwir@ncsu.edu, hp@csc.ncsu.edu

Bushra Anjum  
Department of Computer Science  
National University of Computer and Emerging Sciences  
Lahore, Pakistan  
Email: bushra.anjum@nu.edu.pk

Abstract—In this paper, we evaluated and compared the QoS behavior of video traffic models for H.264 AVC video. The H.264 AVC models that we evaluated are: the Markov Modulated Gamma (MMG) model, the Discrete Autoregressive (DAR) model, the second order Autoregressive AR(2) model, and a wavelet-based model. These models were used to generate synthetic packet traces which were used in a simulation model to estimate the 95th percentile of the end-to-end delay, jitter and packet loss. The QoS metrics of the generated traces are compared with those of the original traces available from the Arizona State University video traces library. We observed that none of the model produces accurate results for every type of video but the MMG model produced the results closest to the actual traces.

I. INTRODUCTION

In the last two decades a plethora of VBR video traffic models have been reported in the literature. These models are stochastic in nature and they are based on analytic and statistical techniques. They are used in performance studies to generate VBR video traffic. An accurate video traffic model should capture the characteristics of a video sequence, such as the distribution of the I, B and P frames and the autocorrelation of successive frame sizes. In addition, the resulting packet trace, when run through an actual network, should behave in terms of QoS metrics, such as, end-to-end delay, jitter and packet loss, as close as the original trace from which the model was developed.

In the literature, the accuracy of a proposed video traffic model is evaluated by testing how close the distributions of the generated I, B and P frame sizes are to those in the original frame trace, which was used to develop the model. This is done using Q-Q plots and the Autocorrelation Function (ACF) of the size of successive frames. These comparisons are visual but they do provide some idea of the model’s accuracy. However, they do not reveal any information about the QoS metrics of the model’s resulting packet trace as the flow of packets is transmitted through a series of network elements such as switches and routers. Since video is transmitted over the IP network, the most important QoS metrics are packet loss, the one-way end-to-end delay and jitter.

In [1], we presented a detailed literature survey of H.264 AVC VBR video models proposed in the last twenty years. The video traffic models were classified in different groups depending on the modeling techniques used. Subsequently, we evaluated four different video traffic models which were representative of the video models proposed in the literature and which are based on the MPEG-4/H.264 standard with GOP patterns that contains I, B and P frames. The evaluation of these four models was based only on obtaining Q-Q plots for the I, B and P frames and comparing the ACF of successive frame sizes generated by the model with that of the original frame trace used to develop the model.

In this paper, we evaluate the QoS behavior of the above mentioned four H.264 AVC video models. Specifically, each model generates a frame trace, which is converted to a packet trace assuming that the video is transmitted over RTP/UDP/IPv4. The packet trace is them offered as input to a tandem queueing network which depicts the path of routers followed by the video stream, and by simulation we estimate the three QoS parameters, namely, the 95th percentile, the jitter, and the packet loss rate. These QoS parameters are then compared against those obtained using the original trace.

This paper is organized as follows: In section II, we describe the four traffic models for H.264 video and in section III, we present our evaluation results. The conclusions are given in section IV.

II. VIDEO TRAFFIC MODELS FOR H.264 AVC VIDEO

In [1], we presented a classification of H.264 video models, of which the following three categories appear to be more popular: (1) Auto-regressive models, (2) Markov-modulated process based models and (3) Wavelet based models.

We implemented two models from the autoregressive group of models, a model from the Markov-based models and a model from the wavelet-based models. Specifically, we implemented the following four models: (1) A Discrete Autoregressive model (DAR) [2], (2) An AR(2) model [3], (3) A Markov-modulated Gamma model (MMG) [4], and (4) A Wavelet model [5].

We chose these models because they are different from each other and also because they are based on MPEG-4/H.264 video with a GOP pattern that contains I, B and P frames. The DAR, MMG and Wavelet based models are fairly recent in their respective categories and are based on H.264 AVC. The DAR model is for video conference applications only and is not suitable for other applications like IPTV or video...
streaming with many scene changes. Therefore, we selected an additional model, i.e., the AR(2) model, from the category of autoregressive models that also incorporates scene changes. Although the AR(2) model was developed for H.261 video, this is the closest to H.264 video as it has three types of frames and a similar GOP pattern.

Below, we describe each model.

A. The Autoregressive models

In an AR process, the current value is a function of a weighted linear combination of past values given by the expression:

$$x(n) = \sum_{i=1}^{p} a_i x(n-i) + e(n),$$

where $a_1, a_2, \ldots, a_p$ are the AR coefficients and $p$ is the order of the AR process. The sequence $e(n)$ consists of i.i.d random variables, known as the residual (or error process), and give the AR its stochastic nature. The residuals are uncorrelated and often assumed normally distributed with zero mean and a variance $\sigma^2$. There are a number of methods to estimate the parameters of the AR process, one of which is the linear prediction.

The AR process is simple as it requires few parameters. When modeling video traffic using the AR process, $x(n)$, represents the bit rate of the $n$th frame or the size in bytes of the $n$th frame of the coded video, $e(n)$ is assumed to be a Gaussian process with zero mean and variance $\sigma^2$. Estimated from an empirical video trace) and lastly $a_i, i = 1, 2, ..., p$, is the lag $i$ autocorrelation of the successive frame sizes.

Below, we describe the two models used in this paper.

1) The DAR(1) Model: A discrete autoregressive model of order $p$, denoted as DAR $(p)$, generates a stationary sequence of discrete random variables with an arbitrary probability distribution and with an autocorrelation structure similar to that of an autoregressive model. A DAR(1) process is a Markov chain with a discrete state space $S$ and a transition matrix: $P = \rho I + (1-\rho)Q$, where $\rho$ is the lag-1 autocorrelation coefficient, and $I$ is the identity matrix. The $Q$ matrix consist of the Pearson type V probabilities $\{f_0, f_1, \ldots, f_k, F_K\}$, where $F_k = \sum_{k>K} f_k$ and $K$ is the peak rate [2]. Each $k$, $k < K$, corresponds to a possible source rate less than the peak rate $K$.

In this model, the frame sizes are expressed in terms of the number of ATM cells per frame. We first obtain the minimum and maximum number of cells per frame, and the mean and the variance for each type of frame, i.e., I, B and P, from the video trace. Using these and the pdf of the Pearson V distribution with parameters $(\alpha, \beta)$, given below, we obtain the rows of the $Q$ matrix:

$$f(x) = \frac{x^{-(\alpha+1)} e^{-\beta/x}}{\beta^{\alpha-1} \gamma(\alpha)}$$  

where

$$\text{Mean} = \frac{\beta}{\alpha - 1}$$

and

$$\text{Variance} = \frac{\beta^2}{(\alpha - 1)^2(\alpha - 2)}$$

The $P$ matrix is then generated using the $Q$ matrix and the lag-1 autocorrelation coefficient $\rho$. From the transition matrix it is evident that if the current frame has $i$ cells, then the next frame will have $i$ cells with probability $\rho+(1-\rho)f_i$, and $k$ cells, $k \neq i$, with probability $(1-\rho)f_k$.

The models starts from a randomly selected state and generates frame sizes while traversing the transition probability matrix until the required number of frames is generated. The I, P and B frames are generated separately using their respective transition probability matrices and then multiplexed according to the required GOP format.

2) A Frame based AR(2) Model: We use a model for MPEG video for all frame types, i.e., I, P and B, incorporating scene changes proposed in [3]. The authors analyzed the ACF and frame size distributions for multiple video traces and showed that the frame size distribution of all three types of frames is lognormal while the scene length distribution can be approximated by a geometric distribution. The scene changes are detected by a considerable change in the size of consecutive I frames. The effect of a scene change on P and B frames is negligible and can be ignored. Therefore, the size of I frame is modeled by two random components: a scene related component and an AR(2) component that accounts for the fluctuations within a scene. The sizes of P and B frames were modeled by i.i.d random processes with lognormal marginal. The complete model is obtained by mixing the three submodels based on the GOP pattern.

The frame-based AR model was implemented as follows. We begin with a trace that contains a sequence of I, B and P frames, and separate the three frame types. Scene changes are determined using the I-frames. Specifically, let $X_I(n), n =1,2,\ldots$, be the the size of $n$th I frame. Suppose we are in the $i$th scene which started with the $k$th I frame. The next I frame, i.e., the $(n+k+1)$th frame belongs to $(i+1)$th scene if

$$\frac{|X_I(n+k+1) - X_I(n+k)|}{\left(\sum_{j=k}^{n+k} X_I(j)\right)/n} > T_1$$

and

$$\frac{|X_I(n+k+2) - X_I(n+k)|}{\left(\sum_{j=k}^{n+k} X_I(j)\right)/n} > T_2$$

where $T_1$ and $T_2$ are two thresholds. We set $T_1=0.05$ and $T_2=0.1$, based on the suggestions in [3]. The scene length is modeled by a geometric distribution, whose parameters are estimated from the trace.

For generating the synthetic trace, we obtain the desired number of scenes from the original trace using the method described above. For each scene, we generate the number of I frames using the above geometric distribution. Then, we sample the first I frame size from a lognormal distribution, and subsequently the P and B frames are generated according
to the GOP format, with frame sizes drawn from a lognormal distribution. The parameters for the three lognormal distributions are determined from the trace.

The next I frame size $X_I(n + 1)$ for the current scene is obtained using the AR(2) process: $X_I(n + 1) = X_I(n) + \Delta_I(n)$, where $\Delta_I(n) = a_1 \Delta_I(n - 1) + a_2 \Delta_I(n - 2) + \varepsilon(n)$, and $\varepsilon(n)$ is a sequence of i.i.d. random variables sampled from the normal distribution with zero mean and a variance $\sigma^2$. $a_1$ and $a_2$ are estimated from the auto-correlation coefficients of $\Delta_I$ at lag 1 and lag 2. $\Delta_I(n)$ is the empirical sequence of differences between the average I frame size per scene and the actual I frame size for the entire trace.

B. A Markov Modulated Gamma (MMG) Model

Models based on a Markov process use states to represent ranges of bit rates or ranges of frames or GOP sizes of a video sequence. Markov processes and Markov chains are often used to modulate other processes such as Bernoulli, Poisson, gamma, and AR. Each state of the Markov process represents a different set of parameters for the particular process. That is, while in a particular state, the model generates values according to the set of parameters associated with that particular state. This is done for a period of time until the process switches to a different state, where it generates values using a different set of parameters. Models of this type are referred to as Markov-modulated. Well-known examples of such models are the Markov-modulated Bernoulli process (MMBP) and the Markov-modulated Poisson process (MMPP).

In [4], a Markov Modulated Gamma (MMG) model was proposed for generating H.264 video traffic. First, a two-pass algorithm is used to partition the video into clips. A clip is a sequence of consecutive $k$ similar sized GOPs. The clips are then organized into shot classes. A shot class of length $k$ is a union of $k$ distinct but not necessarily consecutive clips. A pre-defined number of shots $n$ is used for generating the synthetic trace. In our implementation, we used $n = 7$ as it was the optimal number of shots based on the results in [4]. The GOP sizes were partitioned into these 7 shots. The successive partitioning boundaries of these shots increase in a geometric progression with $a$ as the first term and $b = ar^n$, as the $(n+1)^{th}$ term where $r = e^{(lnb-lna)/n}$, $a$ and $b$ are the GOP sizes corresponding to the 1 and 99 percentile points.

A transition probability matrix is formed for the transition probabilities between the different shot classes. These shot classes are the states of the underlying Markov chain. The transition probabilities are computed from normalized relative frequency of transitions among shot classes as one sequentially traverses all GOPs in the original video i.e.,

$$P_{ij} = \text{Prob}[\text{the next GOP belongs to shot } j \mid \text{the current GOP belongs to shot } i]$$

All frames are partitioned into $3n$ data sets as each shot is sub-partitioned based on type of frame I, B or P. Each of the $3n$ data sets fits an axis-shifted gamma distribution, whose parameters are estimated from the data set it models. Therefore, for 7 shot classes there are 21 gamma distributions for each frame type I, B and P. For the shift we ignore 1% of the data points (frame sizes) and set the value of the shift at the one percentile.

For the generation of a synthetic trace, we start from a randomly selected state, and generate a GOP. The I, P and B frame sizes are sampled from a gamma distribution with their respective mean and variance. After generating all the frames in the GOP, we determine the next state using the state transition matrix. The process is repeated until the desired number of frames is generated.

C. A Wavelet model

Recently, techniques using the wavelet transform have been used to model video traffic. Wavelet analysis is typically based on a decomposition of the signal using a family of basis functions. This includes a high-pass wavelet function that generates the detailed coefficients and a low-pass scaling filter which produces the approximation coefficients of the original signal. The wavelet basis functions absorb the long-range and short-range dependencies by differencing the averages at all time scales, hence the wavelet coefficients are short-range dependent. This makes it possible to model wavelet coefficients as independent (or low-order Markov dependent) random variables without losing much information [6]. Wavelet models for generating synthetic traffic have two parts. First, the coefficients are obtained by applying the wavelet transform to the trace and parameters are estimated for the wavelet correlation model. In the second step, the coefficients are generated using the correlation model and an inverse wavelet transform is applied to get the synthetic trace.

Dai et al [5] proposed a frame-based hybrid framework for modeling MPEG-4 and H.264 AVC and SVC video traffic. They used Haar wavelets to model the distribution of the I frame size and a simple time-domain model for the P and B frame sizes. The detailed coefficients are estimated using a mixture-Laplacian distribution while the coarsest approximation coefficients are modeled as dependent random variables with marginal gamma distribution. Using the estimated approximation and detailed coefficients, the inverse wavelet transform is performed to generate synthetic I frame sizes. It was noted that there is a strong correlation between the P/B frame sizes and the I frame size belonging to the same GOP, called intra-GOP correlation. A linear model that uses this intra-GOP correlation was proposed to generate the P and B frames.

For implementing the wavelet based model, we used the Haar wavelet transform and the following algorithm proposed in [5]:

1) Generate the I-trace
   a) Perform $J$ levels of decomposition on the original I trace
   b) For $i = 1$ to $J$
      i) Estimate the mixture-Laplacian parameters from the original detailed coefficients;
      ii) Generate synthetic detailed coefficients using the estimated parameters.
c) At level J:
   i) Estimate the gamma distribution parameters from the original approximation coefficients;
   ii) Use copula \(^1\) to generate correlated synthetic approximation coefficients.

2) Generate P-traces:
   a) Estimate the parameters of the generalized gamma distribution from the original residual process;
   b) Generate synthetic P-trace based on synthetic I-trace

3) Generate B traces: repeat step 2 using B frames.

III. EVALUATION OF THE FOUR H.264 AVC TRAFFIC MODELS

In this section, we present an evaluation and comparison of the four H.264 AVC traffic models described in the previous section. We used the four different video traces shown in table I. These traces were downloaded from Arizona State University’s video traces library [7], [8] (http://trace.eas.asu.edu/tracemain.html). The GOP size used is 16 with 3 B frames between I and P frames with the following pattern: IBBBPBBFBBBPBBB. We selected these particular traces because we wanted to represent both video conference and IPTV. The NBC news is closer to a video conference while the Star Wars, Tokyo Olympics and the Silence of the Lambs are similar to IPTV programs with many scene changes.

For each model we calculated the distribution of frame sizes and the ACF of successive frame sizes. These results can be found in [1]. Based on the Q-Q and ACF plots for all four video traces, it appears that the MMG model and the wavelet model are good for all types of videos. The frame sizes sequences generated by both these models are very close to the actual traces. The DAR model has an acceptable performance for videoconference type videos only. The AR(2) model generates videos which are not close to any of the actual video sequence.

The QoS behavior of these four models was evaluated using the generated packet trace in a simulation model of the queueing network shown in figure 1. Specifically, each model generates a frame sequence, which is converted into a sequence of IP packets. The H.264/AVC standard introduced the concept of network abstraction layer units (NALUs) which encapsulate the video frames. Each frame is preceded by a prefix NALU of 8 bytes. We assume that the video stream is transmitted using RTP and IPv4, and the maximum transfer unit (MTU) is 1500 bytes. The sender fragments NALUs larger than 1460 bytes to avoid fragmentation along the path. Each packet consists of the 12 byte RTP header, an 8 byte UDP header, and 20 byte IPv4 header. The packet trace is a sequence of pairs of the arrival time of a packet and its packet length.

Video packets are typically transported over a premium IP network that uses a QoS architecture such as MPLS or DiffServ with or without MPLS. In either case, a tandem queueing network consisting of several queues linked in series suffices to study the performance of the video models. In view of this, the queueing model used in this paper, consists of a sequence of 5 single-server queues, with each queue representing the queueing encountered by the packets of the video flow at the output port of each router along the path of the flow. The service time at each queue \( \mu \) depicts the bandwidth allocated to the flow, and the amount of time it takes to serve a packet is equal to its (packet length)/\( \mu \). \( \mu \) is same for all the servers in the queueing network since the same bandwidth is allocated to each link. The mean packet arrival rate \( \lambda \) is calculated from the packet trace by taking the number of bits per second for each second and then taking the average over the period of the entire trace. Each packet preserves its packet length throughout the queueing network.

We fitted each model to each of the four traces listed in table 1, and generated a trace from the fitted models. Each generated trace was then used in the simulation model to estimate the 95th percentile of the end-to-end delay, jitter, and packet loss, as a function of the service rate \( \mu \) at each queue. We note that jitter was calculated as the average of the differences of the end-to-end delays of successive packets. For comparison purposes, we also obtained similar results using the original traces.

The results are given in figures 2 to 5. The traffic intensity is a measure of the average occupancy of the server during a specified period of time, and it is defined as \( \lambda/\mu \), i.e., the mean arrival rate divided by the mean service rate or the bandwidth allocated. Figure 2 gives the traffic intensities and figures 3 to 5 give the 95th percentile of the end-to-end delay, the jitter and the packet loss respectively. Each figure consists of four sets of curves, each corresponding to one of the four traces listed in table I. Each set of curves consists of 5 curves, one for the original trace and one for each model. Figures 2 to 4 were obtained assuming the buffer size of each queue in the queueing network is infinite, i.e., no packet loss can occur. Figure 5 was obtained assuming finite buffers, as will be explained below. We note that the simulation confidence intervals were not plotted as they are very small and they were not discernible.

We can observe in figure 2, that the DAR model generates traffic with a traffic intensity which is the closest to the original trace, the MMG and the wavelet models over-estimate the traffic intensity and the AR2 model under-estimates it. This implies that the average number of bits per second generated by the these models, with the exception of the DAR model, does not match that of the original trace. This is consistent for all four traces.

In figure 3, we see that the MMG model is the closest to the original trace, with the exception of the Silence of the Lambs trace. All other models greatly under-estimate the 95th

\(^1\)Copulas are functions that describe dependencies among variables, and provide a way to create distributions to model correlated multivariate data.
TABLE I: Video Traces used for comparison

<table>
<thead>
<tr>
<th>Movie</th>
<th>Frame Rate</th>
<th>Total number of frames</th>
<th>Movie length (min)</th>
<th>Mean Frame Size (bytes)</th>
<th>Mean GOP Size (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars IV</td>
<td>30</td>
<td>53,997</td>
<td>30</td>
<td>1064</td>
<td>17022</td>
</tr>
<tr>
<td>NBC 12 News</td>
<td>30</td>
<td>49,523</td>
<td>30</td>
<td>1829</td>
<td>29262</td>
</tr>
<tr>
<td>Silence of the Lambs</td>
<td>30</td>
<td>53,997</td>
<td>30</td>
<td>992</td>
<td>15869</td>
</tr>
<tr>
<td>Tokyo Olympics</td>
<td>30</td>
<td>133,125</td>
<td>74</td>
<td>1274</td>
<td>20392</td>
</tr>
</tbody>
</table>

Fig. 2: Traffic Intensity: (a) Star Wars, (b) Silence of the Lambs, (c) NBC News, (d) Tokyo Olympics

percentile of the end-to-end delay. This is probably due to the fact that the original traces are burstier than those generated by the models, see table II. We observe that the Silence of the Lambs trace is much more bursty than the other traces. The mean value, in table II, is the average packet inter-arrival time in milliseconds and the CoV is the coefficient of variation of the packet inter-arrival time. The CoV of the packet inter-arrival time is a measure of the burstiness of the traffic and is defined as the ratio of the standard deviation of packet inter-arrival times to the mean of packet inter-arrival times. A large value for the CoV indicates bursty traffic. We can infer that all models fail to accurately capture the high traffic variability in the original traces. The MMG comes close to the original trace because of its higher arrival rate and larger packet sizes that cause higher delays which in turn causes the 95th percentile delay to increase.

The jitter curves are shown in figure 4. It can be seen that all the models predict jitter fairly accurately within reasonable bounds.

In order to calculate the packet loss, we varied the buffer sizes for a fixed value of $\mu$ which gives a traffic intensity of 0.5 for the original trace. The traffic intensities of the model traces
Fig. 3: 95\textsuperscript{th} percentile of the Delay: (a) Star Wars, (b) Silence of the Lambs, (c) NBC News, (d) Tokyo Olympics

<table>
<thead>
<tr>
<th>Movie</th>
<th>Original Trace Mean</th>
<th>CoV</th>
<th>MMG Mean</th>
<th>CoV</th>
<th>DAR Mean</th>
<th>CoV</th>
<th>AR(2) Mean</th>
<th>CoV</th>
<th>Wavelet Mean</th>
<th>CoV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars IV</td>
<td>23.29</td>
<td>0.99</td>
<td>21.02</td>
<td>0.524</td>
<td>23.54</td>
<td>0.565</td>
<td>23.87</td>
<td>0.427</td>
<td>23.35</td>
<td>0.725</td>
</tr>
<tr>
<td>NBC 12 News</td>
<td>17.89</td>
<td>0.99</td>
<td>15.29</td>
<td>0.543</td>
<td>18.22</td>
<td>0.713</td>
<td>19.46</td>
<td>0.561</td>
<td>18.04</td>
<td>0.664</td>
</tr>
<tr>
<td>Silence of the Lambs</td>
<td>23.38</td>
<td>3.40</td>
<td>22.39</td>
<td>1.10</td>
<td>24.61</td>
<td>0.585</td>
<td>24.21</td>
<td>0.664</td>
<td>22.14</td>
<td>0.645</td>
</tr>
<tr>
<td>Tokyo Olympics</td>
<td>21.79</td>
<td>0.803</td>
<td>18.56</td>
<td>0.552</td>
<td>22.34</td>
<td>0.492</td>
<td>21.68</td>
<td>0.614</td>
<td>21.73</td>
<td>0.562</td>
</tr>
</tbody>
</table>

TABLE II: Trace Statistics: mean and coefficient of variation of packet inter-arrival time (msec)
might be different than the original trace for the same fixed value of $\mu$. All queues were assumed to have the same buffer size. The results are shown in figure 5. We observe that for the Star Wars and NBC News the MMG model over-estimates the packet loss for small buffer sizes and it tends to under-estimate it as the buffer sizes increase. For the Silence of the Lambs movie, the original trace has much higher packet loss than that generated by all four models due to the fact that it generates the most bursty traffic, but for the Tokyo Olympics trace the MMG model over-estimates the packet loss for all buffer sizes. We note that the Tokyo Olympics trace is the least bursty of all traces and MMG generates a trace with a much higher arrival rate than the original (see figure 2d and table II). The rest of the three models produce minimal or zero packet losses. This is due to the fact that the original traces are burstier than the ones generated by the four models. Once again, we observe that all of the models fail to accurately capture the high traffic variability in the original traces. The MMG model produces traffic that is less bursty than the original but it generates larger and more number of packets than the original trace.

Based on the above results it is obvious that there is no perfect model. The accuracy of each model changes with the video type depending on the activity level within the video. Overall, the MMG model has better Q-Q plots and ACF and its QoS behavior is closest to original traces.

**IV. CONCLUSION**

Accurate video traffic models are necessary for evaluating the performance of a new network design, test the performance of an existing network, and evaluate call admission control and bandwidth allocation schemes for video streams. In this paper we evaluated and compared the QoS behavior of four models for H.264 AVC video for different traces representing different types of videos. These models were: the Markov Modulated Gamma (MMG) model, the Discrete Autoregressive (DAR) model, the second order Autoregressive AR(2) model, and a
wavelet-based model. Based on the Q-Q and ACF plots for all four video traces, see [1], it appears that the MMG model and the wavelet model are good for all types of videos. The frame sizes sequences generated by both these models are very close to the actual traces. The DAR model has an acceptable performance for videoconference type videos only. The AR(2) model generates videos which are not close to any of the actual video sequence. However, from the QoS evaluation of these models given in this paper, we can conclude that none of the models produces accurate results for every type of video. There does not appear to be a clear winner, but the MMG model appears to have a better accuracy than the other models.

REFERENCES


