Modeling VP8/WebM Encoded Video Traces

Abdel-Karim Al-Tamimi
Computer Engineering Department
Yarmouk University
Irbid, Jordan
altamimi@yu.edu.jo

***Abstract*— In this paper, we present our research results in modeling video traces encoded with VP8/WebM codec. Our research results are based on our collection of more than 800 VP8 encoded video traces. We show in this paper that our simplified seasonal ARIMA (SAM) model provides a valid model for WebM encoded video traces regardless of their motion, texture levels, or encoding settings. Additionally, we compare the goodness-of-fit of SAM model against simple autoregressive (AR) and automatic ARIMA modeling methods using both visual and statistical tests. Our results show the validity of SAM model as a VP8 video traces model, and its superiority to the other compared models. We conclude this paper with a discussion of the implications of our findings on related areas of research.**

***Keywords*— Modeling, Workload Modeling and Characterization, Video Traces, WebM, VP8, Seasonal ARIMA, SAM model, YouTube.**

I. INTRODUCTION

There has been a fast growing interest inside the multimedia research community in VP8 codec since Google acquisition of VP8 video codec maker, On2 Technologies, in the beginning of 2010. The acquisition was motivated to overcome the anticipated licensing issues that Google might face with their enormous video hosting website, namely YouTube, and to provide a new video standard for HTML5 standard [1]. This intention has been validated recently with the conversion of more than third of YouTube video contents to VP8 codec [2]. Since YouTube is considered as a predominant video streaming website [3], it is forecasted that this move will encourage other video streaming websites and services to take the same step. Google has rebranded VP8 codec as WebM project, and it has included the support of WebP lossy image compression based on WebM technology [4].

With the continuous increase of video streaming share of Internet traffic represented by the 88% increase last year, YouTube alone is responsible for 57% of video streaming traffic [3], there is a pressing need to better understand, model and accommodate the high requirements of video streaming traffic over the Internet. Such models are necessary to provide accurate simulation of video traces behavior and performance over different network topologies and protocols.

VP8, as a new codec, is considered as an exciting topic to both network and video researchers. VP8 codec needs to be thoroughly researched to provide a comprehensive comparison between it and the currently used video codec standards. There have been recent research articles that investigated the difference between VP8 codec and the *de facto* standard of high definition videos, AVC/H.264 codec, in terms of their network performance [5], subjective quality [6], and their encoding characteristics [7,8].

In [5], the authors have conducted a comparison between VP8 and H.264 SVC codecs. In this comparison, both video traffic and video quality metrics have been compared. This paper's calculations and comparisons were based on three video traces: Sony Demo, Die Hard, and Terminator video traces.

In [6], the authors compared the performance of VP8 against AVC/H.264 standard to find that their subjective quality were competitive. In [7,8] the authors compared VP8 to AVC codec in terms of their objective picture quality, where it showed that AVC is slightly better in high motion videos, but provides overall comparable results to AVC codecs. These comparisons considered the fact that VP8 has not been optimized for objective quality measures as some other codecs have.

In network engineering, network simulators are used vigorously to verify and test new Internet protocols. Network simulations are based on either actual video traces, or mathematical traffic-models. Trace-based simulations are prone to many shortcomings concerning: the length of the used video trace compared to the required simulation length, the representativeness of the selected video trace(s), and the limited number of available video traces [9].

For these limitations, there is a need to introduce a valid video traffic model to facilitate future researches in network traffic engineering and other related topics. Such models are preferred to model video traces based on video's elementary stream that represent the video frames independently from video encapsulations and transmission protocols [10].

In [11], the authors performed a modeling analysis study for more than 50 HD video traces encoded with AVC/H.264 codec. The authors compared the modeling and the prediction accuracy of SAM or simplified seasonal autoregressive integrated moving-average (ARIMA) model. In their presented results, they showed the accuracy gain of using SAM model in modeling and predicting videos encoded with AVC/H.264 codec.

978-0-7695-4926-2/12 $26.00 © 2012 IEEE
DOI 10.1109/EMS.2012.82

Similarly, in this paper we discuss our results in verifying the validity of SAM model to represent VP8 traffic based on a large video collection. In the next section, we describe the steps taken to collect our video traces collection that consists of more than 800 video traces. Section 3 discusses SAM model and its characteristics. Section 4 shows our results of the verification steps taken to validate and compare the goodness-of-fit of SAM model. Section 5 concludes this paper with a discussion of the implications of our findings and summarizes the research results.

II. Video Traces Library

In order to provide a comprehensive analysis of the efficiency of a modeling approach, a large collection of video traces needs to be gathered. Compiling the video traces library is a significant task that needs to consider various impacting factors that resides in video content. Thus, the selected videos types need to represent the most common video streams genres available over the Internet. In addition, the researchers need to select the most appropriate encoding settings to help determine the differences among the various codecs.

To ensure a reliable comparison among the video traces, all the videos traces need to be encoded with the same encoding settings. In order to achieve that, all the selected videos need to be converted first to raw video format (YUV420). The conversion process is considerably a long process and requires a significant number of processing and storage recourses [10, 11]. The second step is to encode the videos using common encoding settings that represent diverse video quality levels. Table 1 shows the encoding setting parameters used for the video library. The chosen settings represent 6 different video quality levels, where *Good1* is the lowest, and *Best* is the highest. Figure 1 illustrates the process of converting the selected videos to the final comparable video traces.

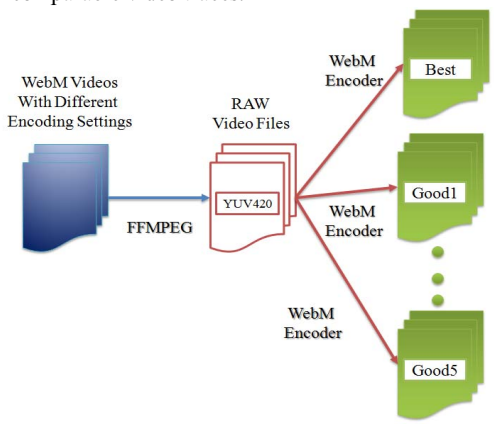

Fig. 1. Video collection main processing steps

As mentioned before, YouTube is considered the predominant website in the video streaming industry, and thus we considered it as the source of our collection of video traces. We collected 10 different video clips from each of the 14 video categories available for YouTube users. Then, the video clips were converted to raw format using FFMPEG [12] library. The conversion assures that all the video traces are encoded with similar encoding settings. The average trace length is 5950 frames, that ranges between 388 and 18029 frames.

TABLE I
Encoding Settings Parameters for The Trace Library

Encoding Profile	Encoding Parameters
Good1	***First pass:*** -p 2 --pass=1 --fpf=tmp.fpf --threads=4 --good --cpu-used=1 --end-usage=0 --auto-alt-ref=1 -v --minsection-pct=5 --maxsection-pct=800 --lag-in-frames=16 --kf-min-dist=0 --kf-max-dist=999999 --token-parts=2 --static-thresh=0 --min-q=0 --max-q=63 ***Second pass:*** -p 2 --pass=2 --fpf=tmp.fpf --threads=4 --good --cpu-used=1 --end-usage=0 --auto-alt-ref=1 -v --minsection-pct=5 --maxsection-pct=800 --lag-in-frames=16 --kf-min-dist=0 --kf-max-dist=999999 --token-parts=2 --static-thresh=0 --min-q=0 --max-q=63
Good2,Good3, Good4, Good5	Same as ***Good1***, but --cpu-used={2,3,4, and 5} respectively
Best	Same as ***Good1***, but --best is used instead of --good without --cpu-used

We have chosen six different encoding settings to better demonstrate the performance of the modeling process against various encoding settings, the selected video encoding settings are shown in Table I similar to the ones used in [6]. We encoded the videos using 720p high definition (HD) resolution. The encoding settings chosen for the six encoding settings are identical except for the *--cpu-used* encoding parameter. This attribute determines the amount of time the encoder spend on each frame to increase the quality of the output video. After encoding the videos, the output files of the encoding process are processed and parsed to extract the video traces at the elementary stream level. In total, we produced 840 video traces. This library of video traces is the basis of our comparison results in this paper.

In the next section, we discuss the mathematical characteristics of SAM model, and the previously achieved results in modeling video traces encoded with different codecs and settings.

III. SAM Model

SAM model is based on Seasonal Autoregressive Integrated Moving-Average (SARIMA) models. These types of mathematical models identify both local and seasonal trends in the observed data traces. SARIMA models are usually represented using the following notation:

$$SARIMA=(p,d,q)\times(P,D,Q)^{s} \qquad (1)$$

where p is the local autoregressive (AR) order, d is the local differencing order, and q is the moving-average (MA) order. P, D, and Q represent the seasonal AR (SAR), the seasonal differencing, and the seasonal MA (SMA) orders, respectively. S represents the seasonality of the date series. Obtaining a seasonal model for data traces consist usually of multiple steps that require human interventions to determine the best data model. For more information, the reader is encouraged to refer to [13, 14].

SAM is a general SARIMA model that has proved its ability to model MPEG4-Part2 [16], H.264/MPEG-Part10 Advanced Video Codec (AVC), and AVC's extension Scalable Video Coding (SVC) video traces [12]. SAM as a general model does not need human intervention to determine the model order, and thus allows its usage in live video applications. SAM as an SARIMA model can be written as:

$$SAM = (1,0,1) \times (1,1,1)^z \quad (2)$$

where z is the seasonality of the video trace. Video trace seasonality depends on the encoding settings of the video, and can be determined easily by inspecting the autocorrelation values between the video successive frames. The main benefits of SAM model over other modeling approaches reside in its simplicity, and versatility to model video traces regardless of their texture and motion levels, and their encoding settings [11].

From (2), we can notice that SAM requires only 4 coefficients to be estimated. The coefficients are: one local autoregressive coefficient (AR) , one seasonal autoregressive coefficient (SAR), one local moving-average coefficient (MA), and one seasonal moving-average coefficient (SMA).

These coefficients are commonly estimated using maximum likelihood (ML) method, conditional sum-of-squares (CSS) method, or a hybrid method where the starting values of the coefficient are estimated using CSS then the estimation process is completed using ML method. In this paper, we will refer to the hybrid approach as (CSS-ML) [14].

Although most literature books suggests ML as the preferred method to estimate SARIMA models [14], it is important to determine the method that provides the best tradeoff between estimation accuracy and computation speed.

In this section, we compare the three estimation methods in their performance, in terms of their estimation accuracy and completion speed, to determine the best suitable method to be used when processing live video streams. Our comparison results are based on averaging the results obtained from running the estimation methods on our collection of 840 VP8-encoded video traces for 10 times.

In the computation performance comparison, and as shown in Table II, CSS has a clear advantage over both ML and CSS-ML in terms of computation speed. On average, CSS method takes only 0.22 seconds to estimate the video trace model, SAM, coefficients. Similarity, the estimation process takes 38.52 seconds using CSS-ML method, and 65.27 seconds using ML method on average.

TABLE II
ESTIMATION METHODS SPEED COMPARISON RESULTS IN SECONDS

Comparison/Method	ML	CSS	CSS-ML
Total execution time (s)	54823.95	186.87	32360.41
Average time per video (s)	65.27	0.22	38.52

For the modeling estimation accuracy comparison, we compared the three methods using three commonly used statistical measures: Mean Absolute Error (MAE), Mean Absolute Relative Error (MARE), and Root Mean Square Error (RMSE) [14]:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|e_i| \quad (3)$$

$$MARE = \frac{1}{N}\sum_{i=1}^{N}\frac{|e_i|}{x_i} \quad (4)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_i)^2} \quad (5)$$

where N is the total number of video frames, e_i is the prediction error calculated at the i-th video frame, and x_i is the i-th video frame size.

As Table III and Table IV show, the difference in accuracy between the fastest estimation method (CSS) and the slowest (ML) is considerably small. In MAE comparison, the biggest difference between the two methods is less than 3% on average. When both compared using MARE measure, the difference is still less than 9%. And when compared using RMSE measure the difference is again less than 3%. Thus, we argue that the resulting degradation of performance due to using CSS method is acceptable when compared to the considerable boost in computation speed it provides compared to the other two methods.

TABLE III
ESTIMATION METHODS ACCURACY COMPARISON RESULTS

Comparison/Method	ML	CSS	CSS-ML
MAE (average)	638.73	657.32	651.05
MARE (average)	0.711	0.772	0.747
RMSE (average)	117012.88	120423.4	117352.13

TABLE IV
PERCENTAGE OF IMPROVEMENT BETWEEN THE ESTIMATION METHODS

Comparison/ Method	ML vs CSS	CSS-ML vs CSS	ML vs CSS-ML
MAE	2.827%	0.954%	1.892%
MARE	8.61%	3.30%	5.14%
RMSE	2.914%	2.617%	0.289%

In this section, we compared three commonly used estimation methods for determining SAM model coefficients. In the next section, we present our results of modeling VP8 video traces using SAM model and the verification steps taken to validate the model accuracy and to compare its goodness-of- fit against other modeling techniques.

IV. VIDEO TRACES MODELING

In our modeling and comparison analysis, we used R project software [17] to model our collection of video traces using SAM model. To validate the goodness-of-fit of SAM model, we computed the empirical cumulative distribution function (ECDF) values of the model trace's frame sizes distribution. Then, we compared the ECDF values of the actual trace to the ECDF values of the model trace.

For N data points (Y_1, Y_2, … , Y_N) with ascending order, the ECDF is defined as:

$$E_N = n(i)/N \quad (6)$$

where *n(i)* is the number of points less than Y_i. Figure 2 shows an example of the visual ECDF comparisons conducted for our video traces collection. As Figure 2 demonstrates, SAM model can be considered as a valid model for the compared actual trace.

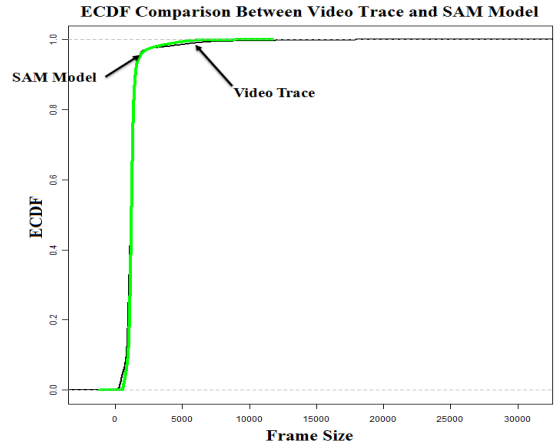

Fig. 2. ECDF comparison between SAM model and actual video trace

To verify SAM model accuracy and to compare it to other regression models, we used the *Kolmogorov-Smirnov* test (K-S test) as a good measure of the goodness of fit. The K-S test indicates if the model data distribution differs from the original video trace distribution using scalar values. We chose K-S test since it is considered as a general nonparametric method, as it makes no assumptions concerning the distribution of the compared data sets [10].

The K-S test compares the ECDF values of the two compared distributions. The K-S test is based on the calculating the maximum difference between the two ECDF curves of the two compared distributions. The null hypothesis (*H0*) assumes that the two distributions are equal. The K-S test produces the maximum difference between the two distributions or K-S statistic, denoted by *D* and it is computed as follows:

$$D = \sup\left|ECDF(X) - ECDF(Y)\right| \quad (7)$$

where *sup* is the *supremum* or the largest absolute difference between the ECDF distributions values. Low *D* values indicate that the model is a good approximation of the actual data and thus an argument in favor of the null hypothesis. Large *D* values indicate different distributions and an argument against the null hypothesis.

Table V summarizes the calculated K-S statistic values (or *D*) for our entire collection of video traces. We can notice that the values of *D* is small on average which supports that the used model, SAM model, is a good fit of the tested video traces.

TABLE V
K-S TEST RESULTS FOR SAM MODELED VIDEO TRACES

K-S Statistic Values				
Encoding Profile	**Maximum**	**Minimum**	**Average**	**Standard Deviation**
Good1	0.512722	0.024302	0.145478	0.096
Good2	0.631931	0.022107	0.194342	0.126976
Good3	0.595519	0.020539	0.178977	0.123027
Good4	0.649355	0.026654	0.138695	0.101707
Good5	0.560126	0.029006	0.138281	0.100139
Best	0.53747	0.026341	0.148981	0.099931
Overall	**0.649355**	**0.020539**	**0.157822**	**0.110603**

Additionally, we compared SAM modeling results with two autoregressive models: simple autoregressive (AR) model [14], and the automatic ARIMA modeling technique (Auto-ARIMA). The Automatic ARIMA modeling returns the best Seasonal ARIMA (SARIMA) model based on the modeling errors. This technique is based on the proposed approach in [17]. Auto-ARIMA is similar to SAM model in its ability to produces a seasonal ARIMA model for the video trace without human interventions. As Table VI shows, SAM provides a 61.18% improvement over the best comparable model on average. Figure 3, shows an example of the visual difference in ECDF comparison values among SAM, AR, Auto-ARIMA models, and the actual video trace.

TABLE VI: K-S TEST RESULTS COMPARISONS BETWEEN AR, AUTO-ARIMA, AND SAM

K-S Statistics Values				
Modeling Technique	**Maximum**	**Minimum**	**Average**	**Standard Deviation**
AR	*0.952324*	*0.035472*	*0.254379*	*0.179267*
ARIMA	*0.932062*	*0.026248*	*0.297602*	*0.220576*
SAM	*0.649355*	*0.020539*	*0.157822*	*0.110603*
Difference %	**27.79%**	**43.63%**	**61.18%**	**62.08%**

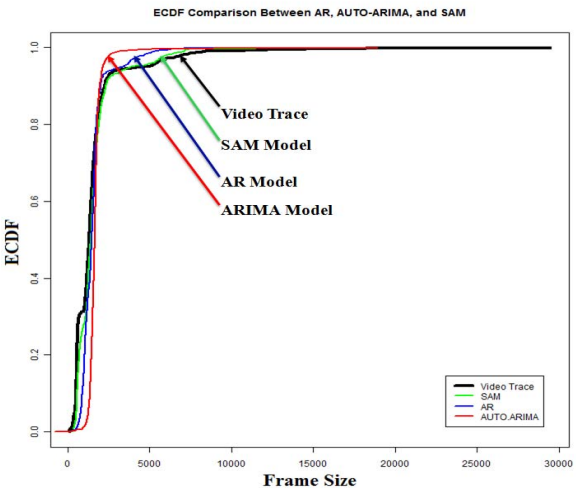

Fig. .3 ECDF comparison among AR, Auto-ARIMA, SAM and actual video trace

As the results in Table VI and Figure 3 show, automatic techniques, like Auto-ARIM, do not always result in better modeling accuracy than the one offered by simple

auto-regression techniques like AR. However, SAM provides a simplified modeling approach without relying on human interventions and without sacrificing the modeling accuracy.

Additionally, to show a visual comparison of the three video traces models, we plotted an actual video trace against its different modeled video traces as shown in Figure 4. As we can notice, SAM has better adaptation to the variations of the video frame sizes, and responds better to the sudden transitions in video frame sizes that usually accompany the beginning of a new video reference frame [11].

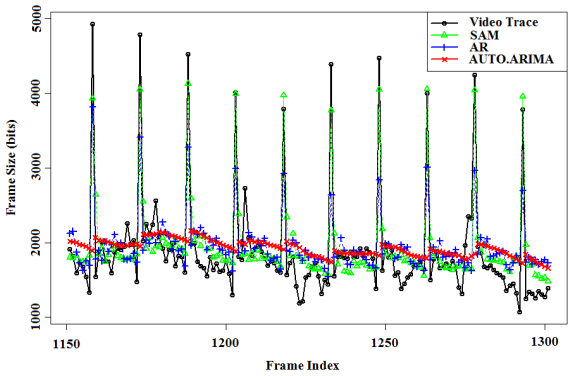

Fig. 4. Video trace modeling comparison between AR, SAM and Auto-ARIMA

V. Conclusions and Future Work

In this paper, we presented our work of analyzing, encoding and modeling of more than 800 video traces that represents a wide variation of video statistical characteristics. Our results show that SAM model is capable of modeling VP8 video traces encoded with various encoding settings, and it is capable of representing VP8 video traces with diverse texture and motion levels.

We compared three commonly used estimation methods for determining SAM model coefficients (CSS, ML, and CSS-ML). Our thorough tests supports our recommendation to use CSS method when modeling live video streams, since the resulting degradation of performance due to using CSS method is acceptable when compared to the significant boost in computation speed it provides compared to the other two methods.

Through both visual and statistical comparisons, we showed that SAM outperforms two commonly used autoregressive models (AR and Auto-ARIMA) , and provides a boost of accuracy up to 61.18% on average using K-S test.

These results show that SAM can present a valid model for VP8/WebM encoded videos, in addition to previously tested codecs: MPEG4-Part2, MPEG4-Part10/AVC, and SVC extension. This conclusion proves the ability of SAM to model most common video traces, and thus improves its importance to be considered to act as a general model and video traffic predictor for more sophisticated traffic engineering algorithms like the ones used in dynamic bandwidth allocation. Our next research step is to determine the applicability of using SAM model in dynamic allocation methods, especially for VP8 traffic, through its video forecasting ability as a time series model.

Acknowledgment

This research was supported by a grant from Yarmouk University (21/2011). The author gratefully acknowledge use if the services and facilities at Yarmouk University.

References

[1] R. Lawler, "Google to Open-source VP8 to HTML5 video" Internet=http://gigaom.com/video/google-to-open-source-vp8-for-html5-video. Apr. 12 2010. [July 2012].

[2] M. Pollice,"Google start converting YouTube Videos to WebM, Opern Source Format," Internet= http://www.brightsideofnews.com/news/2011/4/22/google-starts-converting-youtube-videos-to-webm2c-open-source-format.aspx. Apr. 22 2011 [July 2012].

[3] Marketing Charts,"Video Streaming Held 42% of Share of Global Mobile Bandwidth in H2 2011" Internet= http://www.marketingcharts.com/direct/video-streaming-held-42-share-of-global-mobile-bandwidth-in-h2-11-21206/.[July 2012]

[4] The WebM Projec, "webm an open web media project", Intenert=http://www.webmproject.org [July 2012]

[5] P. Seeling, F. H. P. Fitzek, G. Ertli, A. Pulipaka, and M. Reisslein.," Video network traffic and quality comparison of VP8 and H.264 SVC," In Proceedings of the 3rd workshop on Mobile video delivery (MoViD '10). ACM, New York, NY, USA, 33-38. DOI=10.1145/1878022.1878031 http://doi.acm.org/10.1145/1878022.1878031

[6] F. De Simone, L. Goldmann, J. Leee, and T. Ebrahimi,"Performance analysis of VP8 image and video compression based on subjective evaluations," 2011. Proc. SPIE 8135, 81350M (2011); doi:10.1117/12.896313

[7] J. Ozer, "First Look: H.264 and VP8 compared", Internet=http://www.streamingmedia.com/articles/editorial/featured-articles/first-look-h.264-and-vp8-compared-67266.aspx. May 2010 [July 2012]

[8] D. Vatolin et al.,"MPEG-4 AVC/H.264 Video Codecs Comparison 2010", Internet= http://compression.ru/video/codec_comparison/h264_2010/vp8_vs_h264.html

[9] A. Al-Tamimi, R. Jain, and C. So-In, "Modeling and Prediction of High Definition Video Traffic: A Real-World Case Study," mmedia, pp.168-173, 2010 Second International Conferences on Advances in Multimedia, 2010

[10] R. Jain, "The Art of Computer Systems Performance Analysis", Wiley Professional Computing 1991, ISBN:9780471503361

[11] A. Al-Tamimi, "Mobile and High Definition Video Streams: Analysis, Modeling, Generation, and Dynamic Resource Allocation", Lap Lambert Academic Publishing 2011. ISBN: 3844327401

[12] FFMPEG Library, Internet: http://ffmpeg.org/. July 2012 [July 2012]

[13] G. Box, G. M. Jenkins, and G. C. Reinsel, "Time series analysis : forecasting and control," J.Wiley, 2008

[14] C. Chatfield, "The Analysis of Time Series: An Introduction," Sixth Edition Chapman & Hall/CRC 2003

[15] A. Al Tamimi, C. So-In, R. Jain, "Modeling and Resource Allocation for Mobile Video over WiMAX Broadband Wireless Networks," IEEE JSAC, Special issue on Wireless Video Trans., Vol.28, NO. 3, April 2010.

[16] The R Project for Statistical Computing, Internet: http://www.r-project.org. April 2006 [July 2012]

[17] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecase packge for R", Journal of Statistical Software. 2008