Efficient Modelling of Traffic and Quality of Scalable Video Coding (SVC) Encoded Streams
Piotr Wydrych, Krzysztof Rusek, and Piotr Chołda

Abstract—In this letter, a new model for generating SVC-like traffic and estimating its quality is introduced. Comprising two independent components — i.e., the traffic generator and the quality estimator — the model can significantly speed up the development of SVC applications in various use-cases by relying on approximated properties of streams instead of real traces. The model is verified based on a set of video streams and the performance of its components is assessed.

Index Terms—H.264/AVC video codec, quality of experience (QoE), scalable video coding (SVC), traffic variability, video coding, video quality, video streaming.

I. INTRODUCTION

The increasing number of demanding customers expect video content providers and network operators to deliver high-quality streams regardless of constantly changing mobile network conditions. High adaptivity of a streaming system may be achieved by using scalable video coding (SVC), defined in an annex to the H.264/AVC specification. SVC makes it possible to encode a video into a single multi-layer stream. Layers differ from each other in frame rate, resolution, or quality. These three scalability types are described in detail in [1]. This scalability allows, among other benefits, the provision of end-user devices with a stream of quality varying over time and depending on the devices’ capabilities and network conditions. Thus, SVC is a promising extension which may be used in current and future video streaming systems.

Due to the usage of layers and hierarchical B-frames in SVC, scalable streams differ from ones encoded with H.264/AVC in structure of groups of pictures (GoPs). Therefore, the properties of SVC streams differ significantly from streams encoded with H.264/AVC. This topic has been studied in depth by Seeling and Reisslein [2]. They have encoded a number of videos using different parameters and provided all the traces and an extensive analysis. Gupta et al. [3] have investigated quality scalability. In their work, a comprehensive comparison of characteristics of streams using two quality scalability types — i.e., coarse grain scalability (CGS) and medium grain scalability (MGS) — has been conducted. The traffic characteristics of SVC streams have been examined by Van der Auwera et al. [4]. They have shown that SVC streams feature high bitrate variability and non-negligible autocorrelation of GoP lengths. Van der Auwera and Reisslein [5] have studied the effects of smoothing of SVC streams and have proved that the variability of live streams may be smoothed by aggregating frames forming one GoP into one data block. The abovementioned studies, however, investigate the streams’ quality by looking only at the peak signal-to-noise ratio instead of the widely-used NTIA General Model (known as VQM, [6]) which is linearly-correlated with the Mean Opinion Score (MOS) scale used in subjective tests of the Quality of Experience (QoE).

In this letter, we propose an innovative approach to joint modelling of traffic patterns and quality of SVC streams. We enhance the investigations into the autocorrelation function to consider more than one layer. Then, we study the correlation between traffic characteristics of video sequences and their objectively measured quality. The result of the analysis is used to set up high-performance tools for automated generation of SVC-like traffic and estimation of streams’ quality.

The remainder of the letter is organized as follows. Section II introduces the components of the model and explains the process of its parameters estimation. The model is verified in Section III. The applicability of the model and its implementation’s performance evaluation are given in Section IV. Finally, Section V concludes the letter and provides ideas on further work and possible extensions.

II. COMPONENTS OF THE MODEL

The proposed model consists of two autonomous and independent components: the traffic generator and the quality estimator.
estimator. These may be run separately or together. The process of estimation of their parameters is presented in Fig. 1.

A. Traffic Generator

The parameters of the first component — i.e., the traffic generator — are estimated from a set of SVC stream samples, each containing a list of lengths (in bytes) of consecutive GoPs for all layers. For best results, the samples have to be representative, i.e., they have to be generated using the same encoding scheme as the video streams in the modelled system and the balance between different genres (news, sports, cartoons etc.) has to be preserved. To model both the probability distribution and the autocorrelation function (ACF), the traffic generator uses the Hidden Markov Model (HMM, [7]). The stream samples are used to compute the parameters of HMM with Gaussian outputs using the expectation-maximization (EM) algorithm. Since the value of log-likelihood computed during HMM EM fitting does not take the autocorrelation into account sufficiently, the best fit is determined in two steps. First, the EM algorithm is executed for all the allowed state counts. Then, the fit with the smallest root mean squared error (RMSE) between its ACF values and ACF of training data is selected as the best one. Moreover, minimum GoP length for each layer is saved.

The traffic generator is configured by the following variables: (a) number of layers, (b) number of HMM states, (c) transition matrix, (d) parameters of emission probability distribution in each state, and (e) lower bound of GoP length in each layer. At runtime, the traffic generator repeats the four following steps until interrupted:

1) the next state is randomized;
2) the state-dependent multivariate Gaussian distribution is sampled;
3) for each layer, the obtained random value is rounded to the nearest integer and ensured to be greater or equal to the layer’s minimum; and
4) the vector of GoP lengths in bytes of all layers is emitted.

B. Quality Estimator

To train the second component — i.e., the quality estimator — representative samples of both uncompressed (original) and compressed streams are needed. The streams are gathered into sequences of lengths allowing VQM to be applied (usually 8 to 10 seconds [6]), and the VQM value is computed for each layer of each sequence. Next, the principal component analysis (PCA, [7]) is used to reduce the dimensionality of the information. The GoP lengths are normalized and concatenated into per-sequence vectors. The principal components are computed and the ones which are not needed to explain a predefined percentage (e.g. 90%) of total variance are discarded.

Then the probability distribution function for each remaining principal component is fit to a univariate Gaussian mixture and transformed to the normal distribution $N(0, 1)$. The same fitting and transformation is performed for distributions of VQM values for each layer. Finally, the normalized vectors are concatenated and their covariance matrix is calculated.

The quality estimator is configured by the following variables: (a) number of layers, (b) buffer length, (c) GoP length normalization parameters (mean and standard deviation) for each layer, (d) number of reduced dimensions after PCA, (e) truncated PCA coefficient matrix, (f) parameters of Gaussian mixture fitted to distribution of each principal component, (g) parameters of Gaussian mixture fitted to distribution of VQM values for each layer, (h) covariance matrix. When started, the quality estimator performs the following steps repeatedly:

1) GoP lengths in bytes for all layers are read;
2) each layer’s GoP length is normalized by subtracting the mean and dividing by standard deviation;
3) normalized GoP lengths are added to the buffer;
4) the buffer vector is multiplied by the truncated PCA coefficient matrix to reduce the dimensionality;
5) each principal component is transformed by calculating the normal inverse CDF of its Gaussian mixture CDF;
6) the multivariate conditional normal distribution defined by the covariance matrix and under the condition determined by the transformed principal components is sampled;
7) each variable of the random sample is transformed by the normal inverse CDF of its Gaussian mixture CDF.

Fig. 2. Accuracy vs. complexity, i.e., goodness of fit of autocorrelation function (ACF) of generated GoP lengths vs. Hidden Markov Model (HMM) states used.

Fig. 3. Probability distribution of GoP lengths.

The Gaussian mixture output distribution was also considered, but it does not model autocorrelation function sufficiently well. Therefore, we skip further investigations based on it.

To prevent model overfitting, a rule-of-thumb that the total number of estimated HMM parameters should not exceed 10% of number of input vectors (i.e., the total number of GoPs in training sets) is assumed. This approach is commonly used, e.g., for linear regression and machine learning.

$^1$The Gaussian mixture output distribution was also considered, but it does not model autocorrelation function sufficiently well. Therefore, we skip further investigations based on it.

$^2$To prevent model overfitting, a rule-of-thumb that the total number of estimated HMM parameters should not exceed 10% of number of input vectors (i.e., the total number of GoPs in training sets) is assumed. This approach is commonly used, e.g., for linear regression and machine learning.

$^3$The normal distribution was selected since the conditional distribution of multivariate normal distribution always exists and is also a multivariate normal distribution [7]. This property is important at runtime.
calculating the Gaussian mixture inverse CDF of its normal CDF; and
8) the vector of transformed variables is returned as the VQM score of all layers of the stream for the buffering time.

III. VERIFICATION

The model was implemented and verified in MATLAB. To gather the training data that resemble the content being streamed by centralized and P2P-based Internet television services, 19 high quality (HD 720p) movies of different lengths (feature films, serial episodes, short films) and genres (thriller, documentary, animation, sci-fi etc.) were downloaded from www.vodo.net. Depending on the case, 400 to 1,200 sequences (covering 10% to 30% of total content length) were randomly selected to train the components. They were encoded by the SVC JSVM reference software codec into three layers: base layer L0 (with higher quantization parameter QP) and enhancement layer (with lower QP) split into two MGS layers L1 and L2. Each sequence contained 15 GoPs with G16B15 pattern (one I frame and 15 B frames per GoP). Both traffic generator and quality estimator were trained using the same sequences. The source code is available at: http://wydrych.net/research/svc-traffic-modelling/.

As can be seen in Fig. 2, providing more sequences leads to a more accurate (in terms of goodness of fit of autocorrelation function) and complex (in terms of number of HMM states) traffic generator model. For further analysis, the parameters trained with 600 samples are used. Further incrementing of the sequence count increases the number of states and the model complexity but does not provide a significant accuracy gain.

The cumulative distribution functions (CDF) of the GoP lengths used for training and generated ones are compared in Fig. 3. In Fig. 4, the goodness of fit of autocorrelation function of GoP lengths is presented. These two figures prove that the traffic generator component is able to approximate the
probability distribution and the autocorrelation perfectly at the same time.

Fig. 5 depicts the goodness of fit of the quality estimator’s parameters using the most intuitively descriptive measure of the stream—i.e., the mean bitrate over some period of time (in this case, over the time the video is buffered for VQM computation). As can be seen from the scatter and contour plots, the model can be successfully fitted to the training set and the generated correlation resembles the training one.

IV. APPLICABILITY AND PERFORMANCE

As presented in Fig. 6, three distinct use-cases have been identified to prove the applicability of the solution: (1) ISPs may use the traffic generator to test their traffic management solutions without copying real streams or re-playing traces; (2) content providers may use the quality estimator to assess the quality of the streams they disseminate; (3) research centers may use both components to simulate the behavior of their solutions at all development and deployment stages.

To assess the performance in a non-MATLAB environment, the components have been implemented in Java (using Apache Commons Math 3 for mathematical operations). As discussed above, three use-cases were considered: (1) the traffic generator wrote GoP lengths to the standard output; (2) the quality estimator read GoP lengths from the standard input and wrote VQM values to the standard output; (3) the coupled traffic generator and quality estimator wrote GoP lengths and VQM values to the standard output. To configure the components, the values obtained in MATLAB during the verification were used. The implementation’s performance was assessed on two computers: a server and a PC. The server was equipped with two quad-core Intel Xeon W5590 CPUs running at 3.33 GHz and 48 GiB of RAM. It was running CentOS 5.9 and Oracle JRE 6. The PC was equipped with one dual-core Intel Core i3 540 CPU running at 3.07 GHz and 4 GiB of RAM. It was running Ubuntu 12.04 and Oracle JRE 7. In each use-case and computer combination, the implementation was run independently 30 times, each time generating 4,050,000 samples (30 days of 25 fps stream). Table I presents two measures of performance: (a) the mean number of GoPs per second generated and (b) the mean speedup, i.e., the ratio of the simulated stream length to the simulator’s working time. As can be seen, the speedup of all components is significant and allows processing of huge amounts of data in a short time.

V. CONCLUSIONS AND FURTHER WORK

The traffic and quality properties of SVC traces mean that modeling of the behavior of the SVC-enabled applications cannot yet be performed on the basis of general traffic models. This letter proves that the traffic properties of SVC streams (smoothed to the GoP level) can be approximated by HMM precisely and efficiently. Using not only the probability distribution, but also the autocorrelation, allows engineers and researchers to feed the network with the traffic SVC-like streams.

The correlation of a stream’s properties (including mean bitrate) and its quality cannot be approximated by a simple model such as linear regression. Therefore, a more sophisticated model is needed. The estimation algorithm proposed in this letter is able to recreate the correlation well.

Since the traffic generator training procedure takes only GoP lengths into account, the model is also appropriate for other SVC scalability types. In further studies, it may be worth checking how increasing the number of layers and mixing scalability types influences the components’ accuracy, complexity, and performance. If needed, the implementation of the quality estimator can additionally be enhanced to return a deterministic multivariate probability density function handle instead of a vector of random values.

ACKNOWLEDGMENTS

This work was partially supported by the Polish National Science Centre under Grant N N517 555239. This research was supported in part by PL-Grid Infrastructure.

REFERENCES