A NON-STATIONARY HIDDEN MARKOV MODEL OF MULTIVIEW VIDEO TRAFFIC

Lorenzo Rossi[‡], *Jacob Chakareski*[†], *Pascal Frossard*[†], *and Stefania Colonnese*[‡]

[‡]INFOCOM Dept., Sapienza Università di Roma, via Eudossiana 18, I-00184 Roma, Italy

[†] Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

ABSTRACT

Multiview video is increasingly getting attention due to emerging applications such as 3DTV and immersive teleconferencing. In this paper, we present a non-stationary Hidden Markov Model (HMM) for characterizing the data rate of compressed multiview content. The states of the model correspond to different video activity levels and exhibit a Poisson state duration distribution. We derive a stable maximum likelihood algorithm for estimating the parameters of our multiview traffic model. Synthetic data generated by the model exhibits statistics that closely match those of actual multiview data. In addition, we demonstrate the high accuracy of the model in two multiview streaming applications by evaluating the frame loss rate of a constrained network buffer fed by actual and synthetic data.

1. INTRODUCTION

Multiview Video Coding (MVC) is a new standard for the joint compression of correlated video sequences (views) representing the same scene recorded simultaneously by multiple cameras [1]. The large amount of data and its high rate variability that are typical of MVC content necessitate accurate network dimensioning and resource allocation for successful deployment of multiview applications. Furthermore, the multiple encoding dependencies between the different views make allocating resources in an MVC system much more complex relative to single view scenarios.

Video traffic modeling is an active research area aimed at characterizing the behavior of compressed video content through statistical models. There is a substantial amount of related work on video traffic modeling ranging from applications in teleconferencing [2] to video streaming [3]. In this regard, different stochastic models such as autoregressive processes [4], Transform Expanded Sample (TES) processes [5], and HMMs [6] have been considered. The proposed models are then typically applied to network dimensioning and provisioning, i.e., as a good aid for efficient and accurate allocation of network resources. For instance, one straightforward application of a video model is for generating a synthetic bitstream that is used afterwards to determine the proper size of a network buffer. Moreover, a video traffic model can also be used for deriving procedures for network call-admission-control [7].

Because of the recent nature of multiview applications, there are still no statistical characterizations of MVC compressed content. The present paper provides the first stochastic model that characterizes the frame size sequence of an MVC compressed variable bit rate (VBR) multiview content. To this end, in Section 2, we design a non-stationarity HMM with a Poisson state duration distribution where the different states of the model represent different activity levels of the video source. Having a non-geometric state occupancy distribution allows us to more accurately model the scene activity duration in video content [11], while the specific choice of Poisson distribution allows our model to have the same complexity as a conventional HMM [8, 9]. In Section 2, we also derive a numerically stable maximum likelihood procedure for estimating the parameters of the proposed model. Then, in Section 3 we demonstrate the high accuracy of our model by comparing the histograms and the autocorrelation functions (acf) of frame size sequences corresponding to real and synthetic multiview data generated by the model. Finally, we also show that the proposed model closely matches the behaviour of an actual multiview source by examining their respective frame loss rate in network buffer constrained MVC streaming applications.

The work of J. Chakareski was supported by the Swiss National Science Foundation under Ambizione grant PZ00P2-126416.

2. MVC SOURCE MODELING

2.1. Description of the video source

An MVC source is composed of different video sequences captured simultaneously by multiple cameras. We denote the number of views as N_{View}. At each time instant, the MVC video sequence is composed of N_{View} pictures that may be transmitted together or just a subset of them, depending on the specific multiview application and the end user's needs. The video is organized into Groups Of Pictures (GOP)s, the length of which is denoted N_{GOP}, so that each GOP comprises $N_f \stackrel{\text{def}}{=} N_{\text{View}} \cdot N_{\text{GOP}}$ pictures. We use the terms frame and picture synonymously. The video is encoded using a fixed quantization step size without any rate control mechanism so that the resulting bitstream is completely VBR and its average bit-rate depends on the scene activity of the content. Our model characterizes the frame size sequence generated by the video source for all views.

2.2. Poisson-Hidden Markov Model (P-HMM)

Due to the fixed quantization step size and the scene variability that is typical of video content, the compressed multiview sequence does not represent a stationary stochastic process. In order to account for this, we model the video scene activity by means of a non-stationary HMM, in which the different states correspond to different levels of video scene activity. A random vector $x[n]$ is emitted in each state, where $x[n] \stackrel{\text{def}}{=} [x_0[n], \ldots, x_{N_f-1}[n]]$ represents the set of frames in the n-th GOP of the compressed multiview content. The non-stationarity of the state sequence is achieved by modeling its state duration time via a probability mass function (*pmf*) different from the geometric distribution that in turn is typical for stationary conventional HMMs. Our choice is supported by the study in [11], where it is noted that the duration of video activity is not well described with a geometric distribution. In our case, we employ a Poisson distribution to model the state occupancy in order

to maintain the same number of parameters (hence complexity) as for conventional HMMs.

Now, let us denote the number of states (*i.e.*, different video activity levels) in our model as N_s and the state transition matrix as Π, where π_{ij} denotes the probability of transition from state i to state j. We impose $\pi_{ii} = 0$ since in our case the self-transition probabilities are governed by a (different) Poisson distribution and are denoted by $d_i[k] \stackrel{\text{def}}{=} \frac{e^{-\lambda_i}\lambda_i^k}{k!}$. Given the current state of the model, say i, a random vector $x[n]$ is generated according to the *pmf* $b_i[x[n]]$. The mass functions $b_i[\cdot]$, $i = 1, \ldots, N_s$, have varying number of bins depending on the specific video frame to be generated (I, P, or B) in order to account better for their different complexities. Finally, π_i denotes the probability of the model being in state i.

The generation of synthetic content according to our model is summarized with the following steps:

1. A state is chosen according to the probability distribution $\pi_1, \pi_2, \ldots, \pi_{N_s}$. Assume state i is selected.
2. A state duration time, say k, is generated by the Poisson distribution conditioned on the current state, i.e., $d_i[k]$
3. The HMM stays in the state i for k time instances
4. k video frame vectors $x[n]$ are generated according to $b_i[\cdot]$
5. A state transition is performed according to Π
6. Go back to step 2

2.3. Parameter Estimation

The stage of parameter estimation is crucial in order to have a model able to describe an actual video source. Since the model is part of the HMM family, we can resort to one of the estimation algorithms employed for such models. In particular, an estimation procedure called Expectation-Maximization (EM) algorithm [12] is widely used for HMMs in order to find the maximum likelihood estimate of the parameter set.

In [10], a version of the EM algorithm for P-HMMs is introduced. Unfortunately, for long data sequences, as in video content, this specific algorithm becomes numerically unstable. For details, see [8, 9]. Therefore, we derive a different EM algorithm for parameter estimation that does not exhibit numerical instability. Our algorithm is in major inspired by the work in [8] and represent its extension to the case of non-stationary hidden state duration. A brief summary of the proposed EM algorithm is presented in the following.

Suppose that we observe a video sequence composed of N GOPs. Let $x_0^{N-1} \stackrel{\text{def}}{=} \{x[n]\}_{n=0}^{N-1}$ denote the observed video traffic and $\Theta \in \mathbf{\Theta}$ the parameter set of our model, where $\mathbf{\Theta}$ is the parameter space and $\Theta \stackrel{\text{def}}{=} \{\Pi, \lambda_1, b_1[x], \pi_1, \ldots, \lambda_{N_s}, b_{N_s}[x], \pi_{N_s}\}$. The EM algorithm comprises two computational steps. The first one is an expectation step that computes the auxiliary likelihood function $Q(\Theta|\Theta^{(m)}) = E\{\log(\text{Prob}\{S, x, \Theta\})|x, \Theta^{(m)}\}$, where $S \in \mathbf{S}$ represents a plausible state sequence and $\Theta^{(m)}$ is the current (m-th) estimate of the parameter set. Then, a maximization step follows that maximizes the likelihood function, i.e.,

$$\Theta^{(m+1)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(m)}). \tag{1}$$

The algorithm iterates between the two steps until convergence of the parameter set $\Theta^{(m)}$ is achieved.

The specific computational steps of our EM algorithm, as applied to P-HMMs, comprise

1. The following forward probabilities are defined:[1]

$$\begin{cases} \alpha_n(i,k) \stackrel{\text{def}}{=} P(s_n = i, \ldots, s_{n+k} = i, s_{n+k+1} \neq i | x_0^n, \Theta^{(m)}) \\ \alpha_n(i) \stackrel{\text{def}}{=} P(s_n = i | x_0^n, \Theta^{(m)}). \end{cases}$$

   These quantities are calculated for $n = 0, \ldots, N-1$ and $k = 0, \ldots, N-n-1$ by a recursive algorithm that is not included here due to space constraints. This algorithm is similar to the one described in [8].

2. The a-posteriori probabilities:

$$\begin{cases} \gamma_n(i,k) & \stackrel{\text{def}}{=} P(s_n = i, \\ & \quad \ldots, s_{n+k} = i, s_{n+k+1} \neq i | x_0^{N-1}, \Theta^{(m)}) \\ \xi_n(i,j,k) & \stackrel{\text{def}}{=} P(s_{n-1} = i,\ s_n = j, \\ & \quad \ldots, s_{n+k} = j, s_{n+k+1} \neq j | x_0^{N-1}, \Theta^{(m)}) \end{cases} \tag{2}$$

   are calculated through a backward iteration similar to the one described in [8].

3. Finally, the parameter set $\Theta^{(m+1)}$ is calculated:

$$\pi_i = \sum_{k=0}^{N-1} \gamma_0(i,k) \tag{3}$$

$$\pi_{ij} = \frac{\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-1} \xi_n(i,j,k)}{\sum_{\substack{j=1 \\ j\neq i}}^{N_s}\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-1} \xi_n(i,j,k)} \tag{4}$$

$$b_i[x] = \frac{\sum_{n=0}^{N-1}\sum_{k=0}^{N-n-1} \gamma_n(i,k)\delta_x^{x[n]}}{\sum_{n=0}^{N-1}\sum_{k=0}^{N-n-1} \gamma_n(i,k)} \tag{5}$$

$$\lambda_i = \frac{\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-2}\sum_{\substack{j=1 \\ j\neq i}}^{N_s} k\, \xi_n(j,i,k) + \sum_{k=0}^{N-1} k\, \gamma_0(i,k)}{\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-2}\sum_{\substack{j=1 \\ j\neq i}}^{N_s} \xi_n(j,i,k) + \sum_{k=0}^{N-1} \gamma_0(i,k)}, \tag{6}$$

where $\delta_x^{x[n]}$ denotes the delta function.

3. MODEL ASSESSMENT

In this section, we examine the performance of our model. The multiview content employed in our experiments represents a concatenation of 5 test sequences (Akko & Kayo, Uli, Ballet, Breakdance, Pantomime) that exhibit different motion characteristics so that the concatenated content exhibits varying levels of video scene activity. The concatenated sequence is encoded using the reference encoder JMVC v.7.0 [13] at three different quantization levels $Q_s = 10, 20, 40$ in order to have three encoded sequences at respectively high, medium, and low quality. We have used the following encoding parameters: $N_{\text{GOP}} = 8$, $N_{\text{View}} = 4$, $N = 1016$. The GOP encoding structure is shown in Fig.1. The assessment is performed by comparing the real sequence to a synthetic one generated by the P-HMM when its parameters are estimated from the actual concatenated video content.

Because of the iterative nature of the EM algorithm employed for estimating the parameter set of our P-HMM, we need an initial solution, i.e., an initial estimate $\Theta^{(0)}$. This quantity is crucial for

[1] Our definitions differ from [10] in order to avoid numerical instability.

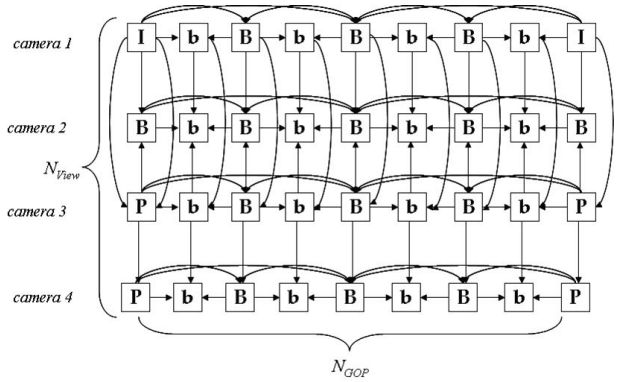

Fig. 1. GOP and encoding structure. The arrows indicate the dependencies between frames.

the proper convergence of the EM algorithm, as otherwise we may end up in a local maximum [8]. We perform the initial estimation stage in two steps. First, we estimate the most likely state sequence of the P-HMM by assigning each GOP of the compressed content to one of the states according to the GOP's average bit-rate. Then, we estimate the P-HMM parameter set by time-averaging the multiview data associated with each state according to the previously estimated state sequence.

3.1. Model validation

We have performed the assessment of our model by comparing the actual multiview sequence and a synthetic one generated according to the model. We compare the two sequences by evaluating the histograms and the autocorrelation functions (acf) of their frame size values. If the model is able to mimic the statistical characteristics of the video source we expect to see the histogram and the acf of the real and the synthetic sequences to be very similar. We expect to observe a larger degree of similarity in the case of the acf, since the acf corresponds to an averaging operation performed over the different histogram bins. Due to space limitations, we have expressed the degree of statistical similarity by means of a percentage error both for the acf and the histogram values. By way of an example, the percentage error for the acf is calculated by the following expression:

$$\textit{percentage error} \stackrel{\text{def}}{=} \frac{\sum_k (\rho_r[k] - \rho_s[k])^2}{\sum_k \rho_r[k]^2} \cdot 100\%,$$

where $\rho_r[k]$ and $\rho_s[k]$ are respectively the acf for the real sequence and the synthetic one. A similar expression is used for calculating the percentage error between the histograms.

The percentage errors for the three sequences are shown in Tab.1. We have calculated the percentage errors by considering both the individual views as well as all views together. First, we would like to remark that most of the percentage errors are under 1%, which means a high degree of similarity between the actual and the synthetic data is achieved. Only for the high quality sequence histograms we have observed a slightly smaller degree of similarity. This is due to the fact that we still employ the same number of histogram bins, as in the cases of low and medium qualities, to sample the frame size values whose dynamic range has increased now due to the finer quantization. In essence, the synthetic data provides a coarser approximation of the frame sizes in this case. Furthermore, as seen from Tab.1 the acf percentage errors are generally lower

	$\mathbf{Q_s = 40}$		$\mathbf{Q_s = 20}$		$\mathbf{Q_s = 10}$	
	Hist	ACF	Hist	ACF	Hist	ACF
View 1	0.6%	0.4%	1%	0.06%	17%	0.5%
View 2	1%	3%	2%	0.03%	8%	0.4%
View 3	3%	3%	1%	0.02%	6%	0.2%
View 4	2%	3%	1%	0.02%	2%	0.2%
All Views	0.5%	0.5%	0.8%	0.03%	2%	0.5%

Table 1. Percentage errors for the acf and histograms of frame size sequences according to real and synthetic data.

than the corresponding histogram percentage errors, as expected and explained earlier. In summary, we can conclude that the proposed P-HMM model is able to accurately represent the statistical characteristics of the actual video source.

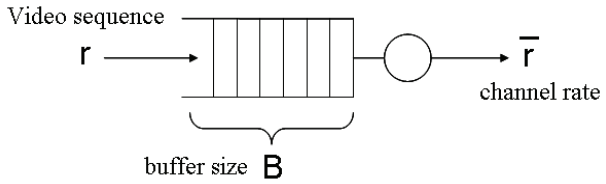

Fig. 2. Buffer filling and depleting.

3.2. Buffer size dimensioning

In this section, we demonstrate that our model is able to reproduce the behaviour of actual video content in the context of streaming. Suppose that the video source is the input of a First In First Out (FIFO) network buffer of finite size B emptied at a constant rate $\bar{r}$, as shown in Fig.2.

One of the resources that should be determined in the stage of network dimensioning, in order to have the desired performance, is the appropriate size of this buffer [2]-[7]. We show now that using the real sequence is equivalent to using the synthetic sequence for determining B. The buffer is fed with real or synthetic data and read out at a constant bit-rate equal to the average bit-rate of the real sequence. Then we compare the two sequences by means of the frame loss rate (an incoming frame is dropped when it is too large to be placed into the buffer) as a function of the buffer size. We perform the test in two different cases: in the first case all the views are transmitted together (typical of a 3DTV-like application), in the second case the user watches a single view but he/she is able to switch among the views at his/her will. In order to have a fair comparison, we suppose that the view switching sequence that indicates the user's requests for view switching, is the same for both sequences (real or synthetic). Specifically, the viewing trajectory starts from view 1 and then switches to view 3, and then to view 2, and finally to view 1.

We remark that in the view-switching case, although the user watches only a single view at a time, he still receives some (or all) frames of the other views because they are needed for decoding the desired view. For this reason, network dimensioning is more difficult in this case and therefore having an accurate source model can be extremely useful. Fig.3 shows the results for the first case and Fig.4 for the second case. In both cases, the synthetic sequences have nearly the same frame loss rate as the real sequences. At high quality, we see a more stepwise shape of the frame loss rate for the

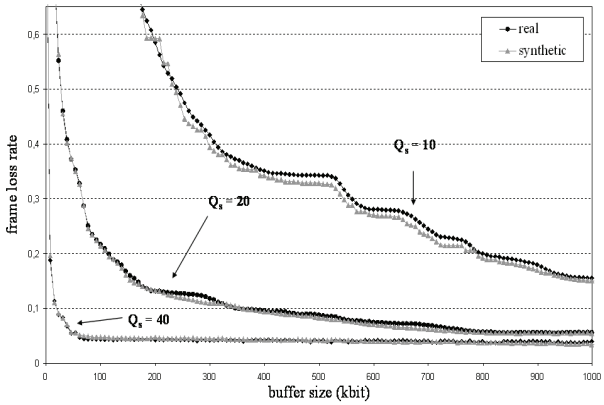

Fig. 3. Frame loss rate for the synthetic sequence (gray - triangle) and the real sequence (black - circle), when all the views are transmitted.

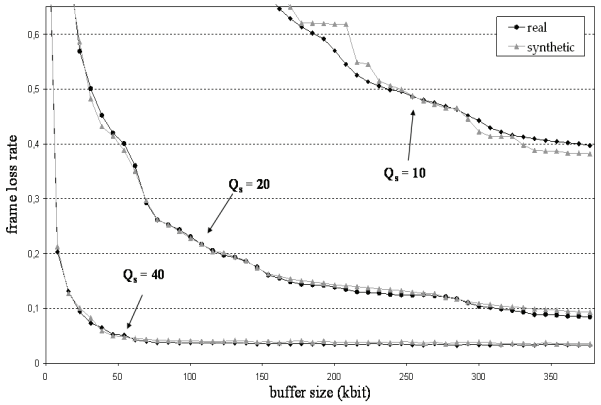

Fig. 4. Frame loss rate for the synthetic sequence (gray - triangle) and the real sequence (black - circle) in the interactive TV case.

synthetic sequence because of the smaller number of active bins of the *pmf* $b_i[\cdot]$. Still, a close resemblance to the frame loss rate of the real sequence is again observed.

Finally, we also examined our model for the network scenario where the buffer size is fixed and equals in size to 1000 ATM cells, while the output bit-rate is varying. Due to space constraints, we only show the results for the view-switching scenario in Fig.5. It can be seen that again the synthetic sequence's frame loss rate matches closely that of the actual video content. Moreover, the bin distribution of the *pmf* $b_i[\cdot]$ has a smaller influence in this network setup, as seen by the very close performances of the syntetic and real sequences in Fig.5 for the high quality case ($Q_s = 10$).

4. CONCLUSIONS

Our work provides the first traffic model of MVC compressed content. To this end, we have designed a non-stationary HMM in which each state corresponds to a different level of video scene activity and the state duration times are modeled with a Poisson distribution. We have derived, for the first time, a numerically stable version of the EM algorithm for estimating the parameters of a non-stationary HMM. Our modeling framework accurately captures the statistical

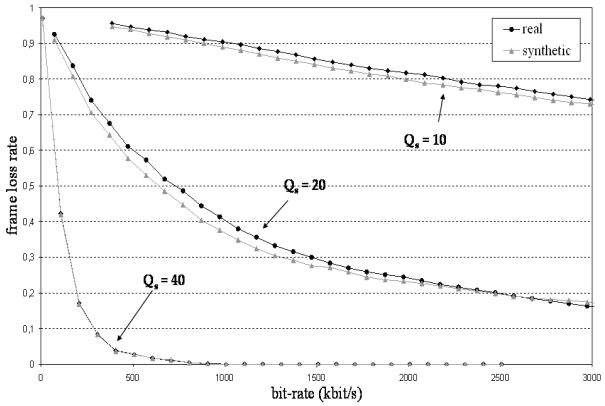

Fig. 5. Frame loss rate for the synthetic sequence (gray - triangle) and the real sequence (black - circle) in the interactive TV case; fixed buffer size case.

properties described by histograms and the autocorrelation function of frame sizes in actual MVC content, both for each of the individual views as well as across all the views together. Furthermore, we have demonstrated that the proposed model closely matches the behavior of a real multiview source in buffer-constrained MVC streaming applications.

5. REFERENCES

[1] Y. Chen, *et al.*, "The Emerging MVC Standard for 3D Video Services, *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 13 pages, 2009.

[2] S. Xu, Z. Huang, "A Gamma Autoregressive Video Model on ATM Networks, *IEEE Trans. on Circ. and Sys. for Video Tech.*, Vol. 8, No. 2, pp. 138-142, 1998.

[3] S. Colonnese, S. Rinauro, L. Rossi, G. Scarano, "Markov Model of H.264 Video Traffic", ISIVC 2008, Bilbao, Spain, July 9-11, 2008.

[4] S. Colonnese, G. Panci, S. Rinauro, G. Scarano, "Markov model of H.264 video sources performing bit-rate switching", *ICIP-2008*, San Diego, USA, October 12-15, 2008.

[5] A. Matrawy *et al.*,, "MPEG4 traffic modeling using the transform expand sample methodology," *IEEE 4th International Workshop on Networked Appliances*, Gaithersburg, 2002.

[6] S. Colonnese, S. Rinauro, L. Rossi, G. Scarano, "H.264 Video Traffic Modeling Via Hidden Markov Process", EUSPICO 2009, Glasgow, UK, 2009.

[7] Z. Zang, *et al.*, "Smoothing, Statistical Multiplexing, and Call Admission Control for Stored Video, *IEEE Jour. on Select. Areas in Commun.*, vol. 15, no. 6, 19 pages, 1997.

[8] Y. Ephraim and N. Merhav, "Hidden Markov processes", *IEEE Trans. Inform. Theory*, vol. 48, pp. 1518-1569, June 2002.

[9] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pages 257-286, 1989.

[10] M. Russell,R. Moore, "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition," ICASSP 85, 1985.

[11] D. P. Heyman, T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic", *IEEE/ACM Trans. on Networking*, vol. 4 no. 1, 1996.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood From Incomplete Data via the EM Algorithm", *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1-38, 1977.

[13] Joint multiview coding (JMVC) v7.0, available via CVS at garcon.ient.rwth-aachen.de.