STOCHASTIC RATE-CONTROL OF INTERFRAME VIDEO CODERS FOR VBR CHANNELS

Julián Cabrera[1] , *José I. Ronda*[2], *Antonio Ortega*[3], *and Narciso García*[4]

[1,2,4]Grupo de Tratamiento de Imágenes
E. T. S. Ingenieros de Telecomuniciación
Universidad Politécnica de Madrid
E-28040 Madrid (Spain)
{jcq,jir,ngs}@gti.ssr.upm.es

[3]Integrated Media Systems Centre
Department of Electrical Engineering-Systems
University of Southern California
Los Angeles, California 90089-2564
ortega@sipi.usc.edu

ABSTRACT

We propose a new algorithm for the real-time control of an interframe video coder operating with a variable rate channel such as wireless channels or the Internet. Using techniques of Stochastic Dynamic Programming we obtain off-line optimal policies from stochastic models of the channel and coder which minimize the average expected distortion. The on-line complexity of our approach is only that required to identify the state of the system (source and channel). The state of the channel is obtained based on the ARQ error-control mechanism, and the source state is computed as complexity measurements on each incoming frame. Simulation results based on this new approach are provided and compared to other proposed rate-control strategies. They show how our model-based optimal policies outperform the other considered approaches keeping a negligible on-line computational cost. This result is very interesting when considering an alternative to traditional costly solutions based on Deterministic Dynamic Programming.

1. INTRODUCTION

The control of a video coder in order to optimize its performance in relation to a given communication channel is a standard problem that has received considerable attention, notably because it is potentially one of the main added values for a concrete implementation of a given standard coder. The importance of rate control and its difficulty have recently increased due to the interest awaken by variable rate channels as those of mobile communications and Internet.

Apart from different heuristic approaches, the systematic treatment starts from the identification of the problem as an optimization issue of the family of dynamic programming problems and the employ of standard tools from this area. Within this general framework the best established approach [7][5] relies on the obtainment of the optimal assignment of operation modes to the different units of the concrete video sequence by solving a deterministic dynamic programming problem. This technique is however somewhat limited in the case of interframe coding, as the amount of data needed to obtain the optimal solutions grows exponentially with the number of frames of the temporal window considered for the optimization, making necessary to resort to different heuristic approximations.

Work partially supported by the Ministerio de Ciencia y Tecnología of the Spanish Goverment under project TIC2002-03692 (DYMAS).

The problem can alternatively be addressed from the point of view of Stochastic Dynamic Programming (SDP) [2]. According to this approach, the system under control (comprising the video coder, the communication channel, and their interfacing elements) is viewed as a stochastic system. Provided that a suitable characterization of this stochastic system is available, it is possible to obtain the optimal control policy for the system in the form of a fixed mapping from system states to control values off-line. The algorithm computes this optimal policy based on the models used for the system characterization. In this way the computational cost of the on-line processing is negligible at the expense of an off-line cost corresponding to the tasks of modeling and control policy computation. The use of this approach for the control of a video coder operating with variable-rate channels has been demonstrated for intraframe coders in [3]. Another work that applies SDP concepts to a similar problem can be found in [4]. The authors address the problem of streaming packetized media (considering packet interdependency) over a lossy packet network. They formulate the problem of minimizing the expected distortion of a finite set of data units where the stochastic factor is due to the channel behavior. The solution to such problem is a vector policy (set of policies, each one for a data unit), where the possible action of each policy at each stage is whether to transmit or not a data unit. Since the media are previously coded and packetized, the R-D values of the data units are available and used in the computation of the vector policy.

In this work we extend the approach proposed in [3] to interframe coding. We formulate an SDP minimization problem where the objective is to minimize the expected distortion of a decoded sequence. The solution in this case consists of a single optimal policy for the whole sequence which indicates the quantizer to use with each frame depending on the state of the system. In our approach we use stochastic models for the video source and the channel. The paper is organized as follows: in Section 2 we formulate the rate control problem; in Section 3 we develop a new stochastic model of the video source that provides the required characterization for the interframe case; in Section 4 we formalize the dynamics of the system; and in Section 5 we show the results obtained and a comparison with rate control strategies based on [8][1].

2. PROBLEM FORMULATION

We tackle the problem of control of a video coder operating in a typical real-time video transmission systems (for either live or stored video), in which we distinguish a video source, a rate control module, a transmission channel, a video coder, and a video decoder with their respective buffers. Real-time operation imposes the need to keep a fixed end-to-end delay, ΔT, between the instant at which the frame enters the system and the instant at which it is presented at the output display. Frames that arrive too late at the decoder are discarded, affecting the distortion of the displayed sequence.

We consider that the transmission channel sends packets of a nominal payload (C) at a constant time interval (T_p). As a consequence of the channel errors (either the packet has been discarded or it has been received with errors), its capacity varies so that the effective payload of the packets is either C (when packets are correctly transmitted) or 0 (if packets are lost). In order to recover from channel errors, we consider that a selective-repeat ARQ mechanism is available in which each packet is acknowledged by the receiver by sending either a positive or a negative acknowledgement (ACK). In the latter case, the transmitter resends the data. The ACK is always received with some delay. We assume that this delay is small compared to the average error burst length, so it can be considered negligible in our problem approach. This assumption ensures the synchronization of the coder and the decoder in the sense of using the same reference frames since the coder knows the transmission result of each frame before encoding the next one. Nevertheless, the proposed formulation would not significantly change in case of having to consider a non-zero feedback delay, although a synchronization mechanism between the coder and the encoder should be considered, such as finishing the transmission of every frame even if it becomes a lost frame or reencoding the frame when the coder knows that the reference frame is lost. A time-out mechanism is established in order to protect from lost ACKs. Because of the end-to-end delay and the limited channel capacity there is an upper bound to the physical size of the coder and decoder buffers. So, we will consider that they are large enough not to impose any further constraint to the system operation. Regarding the transmission channel, we assume a first order Markov model [9] in which the channel switches between states s_0 and s_1. In state s_1 packets are received correctly while in state s_0 packets are received with errors. We concentrate here on H.263 encoding scheme [6], using interframe coding in the following manner: (i) the first frame is intra-frame coded, (ii) the rest of the sequence is coded using P frames. We apply the same quantizer to a whole frame.

The real-time video transmission system is modeled as a discrete-time dynamic stochastic system in which each stage (process of coding and transmitting a frame) is characterized by the following state variables:

- Complexity of the current frame: a characteristic of the frame to be coded which summarizes its rate-distortion properties, as described in the following section.
- Channel state: variable that indicates whether the last packet was correctly transmitted or not.
- Time left: number of channel slots available for packet transmission before the due time of the current frame.
- Previous quantizer, used with the previous frame.

The goal of our rate control is the selection of allowable actions (quantizers q_k) within the set Q of available quantizers defined in the H.263 standard. As a result of the selected action and the channel state evolution, the decoded frame will show some distortion, which we consider to be the cost to be minimized. We define this distortion as the mean square error (MSE) of the displayed frame with respect to the original one. The distortion of the whole sequence is then:

$$D_{sequence} = \sum_{k \in \mathcal{T}} d_k + \sum_{m \in \mathcal{L}} d_{lf,m}, \tag{1}$$

where d_k indicates the distortion due to the encoding of frame k, $\mathcal{T}$ is the set of indices of the correctly received frames , $d_{lf,m}$ indicates the distortion due to the loss of frame m, and $\mathcal{L}$ is the set of indices corresponding to lost frames.

With our approach the on-line computational cost required is only that of identifying the state of the system: the frame complexity is computed as a rate measurement on the current frame, the channel state and time left is based on the ARQ mechanism.

3. SOURCE MODEL

The use of SDP techniques requires the modeling of the system as a *controlled Markov chain*, i.e., as a stochastic system in which the state probabilities at time $k+1$ only depend on the state at time k and the control parameter q_k. Following the general structure for the video source model presented in [3], we propose the next model which covers two different aspects: (i) R-D modeling within a frame (frame model), and (ii) modeling its evolution from frame to frame (sequence model). The final objective of the proposed model is to estimate accurately all the R-D points of a frame (coded in interframe coding) from computed measurements, namely the *frame complexity*. This can be achieved exploiting the relationship of the R-D values present at different frames in a similar way to that presented in our previous work [3].

3.1. Frame model

The frame model aims at estimating the R-D values of a frame k at any quantizer q_k ($r_k(q_k)$ for rate and $d_k(q_k)$ for distortion) from two reference R-D values at quantizers q_r for rate and q_d for distortion, as it was done in [3]. The equations for this part of the model are:

$$\begin{aligned} r_k(q_k) &= a(q_k, q_r)\, r_k(q_r) + b(q_k, q_r), \\ d_k(q_k) &= c(q_k, q_d)\, d_k(q_d) + d(q_k, q_d). \end{aligned} \tag{2}$$

Note that the form of these equations is more general than that of their counterparts in [3].

In order to assess the accuracy of the frame model we have computed several instantiations for different reference quantizers of the sequence "silent" measuring the mean relative error of the estimates. Table 1 shows the results.

Although the level of accuracy of this model is lower than that of the frame model for the intraframe case (see [3]), it still provides a reasonable degree of approximation, which provides competitive control policies as Section 5 shows.

3.2. Sequence Model

The sequence model characterizes the evolution of the R-D curves through the reference values of the frame model ($r_k(q_r)$, $d_k(q_d)$). This evolution depends on the characteristics of the frames and, due to the use of interframe coding, on previous quantizers. For frame k, we model this new dependency through the previous quantizer q_{k-1} in the following way:

$$r_k(q_r, q_{k-1}) = e(q_{k-1}, q_z)\, r_k(q_r, q_z) + f(q_{k-1}, q_z),$$
$$d_k(q_d, q_{k-1}) = g(q_{k-1}, q_p)\, d_k(q_d, q_p) + h(q_{k-1}, q_p). \quad (3)$$

Note that q_r and q_d are the reference quantizers of the frame model, while q_z and q_p are the reference quantizers of the sequence model.

The accuracy of the previous equations is shown in Table 1. We have computed several instantiations for different reference quantizers of the sequence "silent" measuring the mean relative error of the estimates.

Therefore, using equation (3) and from measurements $r_k(q_r, q_z)$ and $d_k(q_d, q_p)$ it is possible to estimate the reference values for the frame model at any other previous quantizer ($r_k(q_r, q_{k-1})$, $d_k(q_d, q_{k-1})$) with a reasonable level of accuracy.

In order to model the dependency the evolution of the R-D characteristics we use variables $r_k(q_r, q_z)$ and $d_k(q_d, q_p)$. Note that using equations (3) from these values we can estimate the reference values for the frame model at any other previous quantizer ($r_k(q_r, q_{k-1})$, $d_k(q_d, q_{k-1})$), and using equations (2) we can estimate any R-D value for frame k at any quantizer. Nevertheless a further simplification is possible since there is a strong correlation between these variables. Therefore we can approximate one variable as a function of the other as follows:

$$d_k(q_d, q_p) = m(q_z, q_p)\, r_k(q_r, q_z) + n(q_z, q_p). \quad (4)$$

Thus, our sequence model only requires the characterization of the evolution of one variable: $r_k(q_r, q_z)$. We model this evolution as a process with uniformly distributed samples: s_k. This simple model provides reasonable results as Section 5 shows.

	Frame		Sequence	
	Rate	Dis.	Rate	Dis.
Err. Min	3.9 %	1.0 %	2.6 %	2.7 %
Err. Max	21.2 %	12.3 %	11.8 %	3.4 %

Table 1. Range of the mean relative error of the source model.

4. SYSTEM DYNAMICS AND OPTIMAL POLICY

Now that the elements of the system have been described, it is necessary to formalize the dynamics of the system. Since our system comprises two stochastic elements, the source and the channel, its dynamics is expressed as the joint transition probabilities for every action. Our goal is thus to obtain the probabilities:

$$P\left[s_{k+1}, c_{k+1}, t_{k+1}, a_{k+1} | s_k, c_k, t_k, a_k, q_k\right]. \quad (5)$$

where the subindex k indicates the stage or frame index and, at that stage, s_k is the complexity of the frame, c_k is the state of the channel, t_k is the time left for the current frame, a_k indicates the previous quantizer (q_{k-1}), and q_k is the selected quantizer. In the remainder of this section, we show the exact computation of the above probabilities. As the s_k is independent of the others variables, and since we have assumed that it does not depend on previous stages, we can rewrite (5) as:

$$P\left[s_{k+1}, c_{k+1}, t_{k+1}, a_{k+1} | s_k, c_k, t_k, a_k, q_k\right]$$
$$= P\left[s_{k+1}|s_k\right] P\left[c_{k+1}, t_{k+1}, a_{k+1} | s_k, c_k, t_k, a_k, q_k\right]. \quad (6)$$

The source model provides the first term of (6), while the second one requires a further elaboration.

As a_{k+1} only depends on q_k we can rewrite it as:

$$P\left[c_{k+1}, t_{k+1}, a_{k+1} | s_k, c_k, t_k, a_k, q_k\right]$$
$$= P\left[a_{k+1}|q_k\right] P\left[c_{k+1}, t_{k+1} | s_k, c_k, t_k, a_k, q_k\right], \quad (7)$$

where $P\left[a_{k+1}|q_k\right]$ is 1 if $a_{k+1} = q_k$ or 0 if $a_{k+1} \neq q_k$.

For the second part of (6) we use again the source model to compute the number of packets that the encoded frame occupies, n_k. Using n_k we can rewrite it as follows:

$$P\left[c_{k+1}, t_{k+1} | s_k, c_k, t_k, a_k, q_k\right] = P\left[c_{k+1}, t_{k+1} | n_k, c_k, t_k\right] \quad (8)$$

Thus, it is only needed to compute the joint probability $P\left[c_{k+1}, t_{k+1} | n_k, c_k, t_k\right]$. The exact computation of such probability can be found in detail in [3]. Table 2 shows a summary of the notation and equations used in its computation.

Using the equations of the dynamics of the system, standard SDP algorithms [2] find optimal stationary policies (i.e., policies that do not depend on the stage), which indicate the appropriate action for every system state: $q_k = \mu(s_k, c_k, t_k, a_k)$. The computation of the policies is carried out off-line, so that during on-line operation only state identification is necessary.

Notation	
s_k	frame k complexity
c_k	channel state
t_k	time left for frame k
a_k	quantizer for frame $k-1$
q_k	quantizer for frame k
n_k	size of coded frame k in packets
w_k	packets devoted to frame k transmission
l_k	first packet slot for frame k
n_{pp}	no. of padding packets
$c_{(l)}$	channel state at packet slot$l$
$v_{(a,b)}$	no. packets correctly transmitted in [a, b]

Transition Probabilities
$P\left[c_{k+1}, t_{k+1} \vert n_k, c_k, t_k\right] = \sum_{w_k} P\left[c_{k+1}, w_k \vert n_k, c_k, t_k\right]$
$P[c_{k+1}, w_k \vert n_k, c_k, t_k]$ $= \sum_{i=0,1} P[c_{(l_k+w_k+n_{pp})} = c_{k+1} \vert c_{(l_k+w_k)} = i]$ $P[c_{(l_k+w_k)} = i, w_k \vert n_k, c_{(l_k-1)} = c_k, t_k]$
$P[c_{(l_k+w_k)} = 1, w_k \vert n_k, c_k, t_k]$ $= P[v_{(l_k,l_k+w_k)} = n_k, c_{(l_k+w_k)} = 1 \vert c_{(l_k-1)} = c_k]$
$P[c_{(l_k+w_k)} = 0, w_k \vert n_k, c_k, t_k]$ $= P[v_{(l_k,l_k+w_k)} = w_k - (t_k - n_k + 1), c_{(l_k+w_k)} = 0 \vert$ $c_{(l_k-1)} = c_k]$
$P[v_{(l_k,l_k+w_k)} = h, c_{(l_k+w_k)} \vert c_{(l_k-1)}]$ $= P[v_{(l_k,l_k+w_k-1)} = h, c_{(l_k+w_k-1)} = 0 \vert c_{(l_k-1)}]$ $P[c_{(l_k+w_k)} \vert c_{(l_k+w_k-1)} = 0]$ $+P[v_{(l_k,l_k+w_k-1)} = h-1, c_{(l_k+w_k-1)} = 1 \vert c_{(l_k-1)}]$ $P[c_{(l_k+w_k)} \vert c_{(l_k+w_k-1)} = 1]$

Table 2. Summary of notation and transition probability eqs.

5. RESULTS AND CONCLUSIONS

In order to assess the performance of the proposed rate control scheme, optimal policies based on our model have been obtained and used in a simulated burst-error transmission environment. For our tests we have used the QCIF sequences "mother-daughter", "akiyo" and "silent" (300 frames long at 30 frames per second) transmitted at 15 frames per second (T_f=1/15 s). The first frame is intraframe coded and the rest of frames are coded as P frames. The channel simulator transmits 328 bit packets every 5 ms ($C = 328$, $T_p = 5$). We have used a model for the channel with an average burst length of 19 packets with the following transition probabilities: $P(s_1|s_0) = 0.0091$, $P(s_0|s_1) = 0.0526$.

In order to get a meaningful representation of the channel behavior in every simulation (experiment) we transmit a sequence 100 times at different starting points in the channel simulator and average the results. The number of frames transmitted in each test adds up to 15,000 frames, and the number of packets used is about 192,000. We have run the simulations with different values of ΔT: 133 and 267 ms ($2T_f$ and $4T_f$). The following parameters have been computed in order to measure the quality of the transmission:

- PSNR of the transmitted sequence. The distortion of a lost frame is computed as the distortion of the frame displayed instead of it (last frame correctly transmitted) with respect to the corresponding current frame.
- PSNR of the correctly transmitted frames (PSNR$_{tx}$).
- Proportion of lost frames (n_{lf}).

In addition to our proposed algorithm (SDP), the tests have been performed with other two algorithms:

- Algorithm based on the TMN8 [8] rate control (TMN). In this case, we have used the strategy proposed for the frame-layer rate control (prediction of the target rate for the current frame). Then, we use the quantizer that produces the closest encoding rate to the predicted value.
- Algorithm based on the strategy proposed in [1] (TMN-MOD). The authors propose an customization of the TMN frame-layer rate control mechanism for variable bit-rate channels similar to those considered in this work. They include in their algorithm an estimation of the average retransmitted bits which is discounted from the target rate. In our implementation we use the quantizer that produces the closest encoding rate to the predicted value.

Results are shown in Table 3. We can note that the SDP algorithm provides remarkable better results than the other two approaches. In terms of PSNR, SDP algorithm provides an increase of the PSNR with respect to TMN-MOD from 1.18 dB to 2.31 dB, and with respect to TMN from 1.37 to 2.57 dB. Regarding the number of lost frames, SDP algorithm achieves a significant lower number of lost frames than TMN-MOD (from 2.4 to 8.3) and TMN (from 3.9 to 12.5). In addition to reducing the number of lost frames SDP provides higher PSNR$_{tx}$ values than the other approaches in most of the cases.

Regarding the on-line computational cost, the three algorithms requires similar resources. In our implementation of TMN and TMN-MOD a variable number of frame coding operations is needed in order to determine the appropriate quantizer. For the SDP algorithm, it is required a maximum of three coding operations for each frame (depending on the selection of the reference quantizers of the source model it can be reduced to only two coding operations), two (or one) operations are required to determine the frame complexity and the other corresponds to the selected quantizer.

Therefore, we can conclude that the SDP algorithm provides an increase in quality of the decoded sequence with respect to the other two considered algorithms, maintaining a reasonable on-line computational cost. This result is of much interest if we consider the computational cost required by algorithms based on Deterministic Dynamic Programming [5]. Due to the interframe dependency, the computational cost associated to these algorithms is a real handicap for their use in real-time [7]. However, by means of SDP techniques the computational requirements of using interframe coding is absorbed in the computation of the optimal policies, process that is carried out off-line. As a consequence, the on-line computational cost required is quite similar to that of the intra-frame coding.

	$\Delta T = 2T_f$			$\Delta T = 4T_f$		
"mother-daughter"						
Algorithm	PSNR	PSNR$_{tx}$	n_{lf}	PSNR	PSNR$_{tx}$	n_{lf}
SDP	34.87	36.49	12.9	36.07	36.97	6.3
TMN-MOD	33.43	36.38	22.5	34.36	35.86	9,8
TMN	33.42	36.46	23.2	33.50	36.40	14.8
"akiyo"						
Algorithm	PSNR	PSNR$_{tx}$	n_{lf}	PSNR	PSNR$_{tx}$	n_{lf}
SDP	37.56	38.74	11.9	38.30	39.05	5.3
TMN-MOD	35.49	36.78	15.8	35.99	36.78	7.9
TMN	35.53	37.24	18.0	36.03	37.19	10.2
"silent"						
Algorithm	PSNR	PSNR$_{tx}$	n_{lf}	PSNR	PSNR$_{tx}$	n_{lf}
SDP	31.26	32.86	13.9	32.46	33.37	6.6
TMN-MOD	30.08	32.38	14.4	30.94	33.34	10.6
TMN	29.89	32.95	26.4	30.13	32.95	19.0

Table 3. Results of the test sequences.

6. REFERENCES

[1] S. Aramvith, I. Pao, M. Sun, "A rate-control scheme for video transport over wireless channels", IEEE Trans. on Circ. and Sys. for Video Tech., Vol.11, No.5, pp. 569-580, May 2001.

[2] D. P. Bertsekas, "Dynamic Programming and Optimal Control", Athena Scientific, USA, 1995.

[3] J. Cabrera, A. Ortega, J. I. Ronda, "Stochastic rate-control of video coders for wireless channels", IEEE Trans. on Circ. and Sys. for Video Tech., Vol.12, No.6, pp. 496-510, June 2002.

[4] P. A. Chou, Z. Miao, "Rate-distortion optimized sender-driven streaming over best-effort networks," IEEE Workshop on Multimedia Signal Proc., Cannes, pp. 587-592, Oct. 2001.

[5] C. Hsu, A. Ortega, M. Khansari "Rate control for robust video transmission over burst-error wireless channels", IEEE Journal on Selected Areas in Comm. Special Issue on Multimedia Network Radios, Vol.17, No.5, pp. 756-773, May 1999.

[6] ITU-T, "Video coding for low bit-rate communications", ITU-T Recommendation H.263; v.1 Nov. 1995, v.2 Jan. 1998.

[7] A. Ortega, K. Ramchandran and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations", IEEE Trans. on Image Proc., Vol.3, No.1, pp. 26-40, Jan. 1994.

[8] J. Ribas-Corbera, S. Lei, "Rate control in DCT video coding for low delay communications", IEEE Trans. on Circ. and Sys. for Video Tech., Vol.9, No.1, pp. 172-185, Feb. 1999.

[9] M. Zorzi, R.R. Rao, y L. Milstein, "On the accuracy of a first-order Markov model for data transmission on fading channels", in Proc. IEEE ICUPC'95, Tokyo, Japan, Nov. 1995.