Predicting Slice Loss Distortion In H.264/AVC Video For Low Complexity Data Prioritization

Seethal Paluri[1], Kashyap K. R. Kambhatla[2,3], Sunil Kumar[2], Barbara Bailey[4], Pamela Cosman[3], John Matyjas[5]

[1] Computational Science Research Center, San Diego State University
[2] Department of Electrical and Computer Engineering, San Diego State University
[3] Department of Electrical and Computer Engineering, University of California, San Diego
[4] Department of Mathematics and Statistics, San Diego State University
[5] Air Force Research Laboratory, Rome, NY
spaluri@sciences.sdsu.edu, kkambhat@ucsd.edu, skumar@mail.sdsu.edu, babailey@sciences.sdsu.edu,
pcosman@ucsd.edu, john.matyjas@rl.af.mil

***Abstract*—We propose a low complexity Generalized Linear Model (GLM) for prioritizing slices during real-time H.264/AVC compressed video streaming. We train the GLM over a video database to predict the Cumulative Mean Square Error (CMSE) corresponding to individual slice losses by using a combination of efficient video parameters which can be easily extracted during the encoding of a frame. We prioritize the slices generated within a Group of Pictures (GOP) based on the predicted CMSE by using a Quartile Based Prioritization (QBP) scheme. For comparison, we also perform QBP on measured CMSE values from individual pre-encoded slice losses and analyze the priority misclassifications of the slices. We validate our model by applying Unequal Error Protection (UEP) using RCPC codes to the different prioritized bitstreams and evaluating their performance over noisy channels. Simulation results show that predicted CMSE schemes achieve PSNR performance close to that of the measured CMSE schemes for different slice sizes, video bitrates and over different channel SNRs.**

***Index Terms*—H.264/AVC video compression, real-time CMSE prediction, slice prioritization, RCPC codes, unequal error protection.**

I. Introduction

The demand for real-time video transmission over wireless networks is increasing rapidly. In order to efficiently utilize limited wireless bandwidth, video data is compressed using sophisticated video coding techniques such as H.264/AVC, which is the state-of-the-art and widely used video coding standard jointly developed by ITU and ISO [1]. As real-time transmission of multimedia content is becoming popular, the study of video quality evaluation and monitoring has also gained importance. Modeling video quality for such systems has two basic requirements: firstly, the model should account for various video application parameters accurately, and secondly, the complexity of parameter calculation should be kept low.

Our work is motivated by past research in [2]–[6] where subjective techniques are used to model the video packet loss visibility. The authors in [2] focused on estimating Mean Square Error (MSE) using three approaches (Full-Parse, Quick-Parse, and No-Parse) to access spatio-temporal parameters. This work was extended to MPEG-2 video bitstreams in [3] using two techniques: (*i*) a tree based classifier called Classification and Regression Trees (CART) [7] that labeled each possible packet loss as being either visible or invisible, and (*ii*) a Generalized Linear Model (GLM) which predicts the probability that a packet loss will be visible to an average viewer. Scene significance characteristics were explored in [4] through packet loss impairments in MPEG-2 and H.264 compressed videos by using Patient Rule Induction Method (PRIM). It was extended in [5] and a versatile GLM was developed that is applicable for different compression standards, concealment techniques and Group of Pictures (GOP) structures by considering attributes of packet loss impairments.

Developing these perceptual loss visibility models using subjective techniques is time consuming as human observers are needed to evaluate a visibility score before parameters can be modeled. Though we share the same motivation to gauge video quality, this paper focuses on evaluating the relative importance of each H.264 video slice, based on its expected loss distortion determined as predicted Cumulative MSE (CMSE) from a low complexity GLM model. By monitoring the video content being transmitted over a wireless network, our low complexity model can be easily updated and re-trained periodically to improve the prediction accuracy over time. We propose a model development framework which determines the subset of video parameters affecting slice loss distortion to predict the CMSE value from a video database used as a training set. Later these parameters are extracted real-time during the encoding process for performing Quartile Based Prioritization (QBP) of slices transmitted over an AWGN channel.

The remainder of the paper is organized as follows. Section II discusses the video factors used to model the impact of a slice loss, followed by the model development in Section III. Section IV discusses our QBP slice prioritization scheme and problem formulation for minimizing the expected video distortion over an AWGN channel by providing unequal error protection (UEP) using Rate Compatible Punctured Codes (RCPC). The simulation setup and experimental results are discussed in Section V. Section VI concludes the paper.

II. Video Factors Affecting Slice Distortion

We study the video factors that capture the effect of individual slice losses on video quality and help in predicting their distortion in terms of CMSE. We consider only those factors that are available during the encoding process by evaluating (*a*) the encoded frame, and (*b*) the error frame at the location of slice loss. Let the original uncompressed video frame at time t be $f(t)$, the reconstructed frame without the slice loss be $\hat{f}(t)$ and the reconstructed frame with the slice loss be $\tilde{f}(t)$.

A. Encoded Frame Factors

The attributes of slice loss can be expressed in terms of the underlying video content. The magnitude of distortion induced by a slice loss is influenced by the presence of texture components, luminance masking and motion masking. For our model, we study the following factors extracted at the location of the slice loss during the encoding process.

- **Motion Characteristics**: We compute the mean motion vectors **MOTX** and **MOTY** over all the Macroblocks (MBs) in the slice.

- **AVGINTERPARTS**: Represents the number of sub-partitions averaged over the total number of MBs in the slice. If the underlying motion is complex, **AVGINTERPARTS** would be high.
- **Maximum Residual Energy (MAXRSENGY)**: First, Residual Energy (**RSENGY**) is computed for a MB as the sum of squares of all its integer transform coefficients after motion compensation. Then **MAXRSENGY** of a slice is equal to the highest **RSENGY** value of all the MBs contained in it. If the scene has high motion, then the MAXRSENGY would also be high.
- **Signal Characteristics**: We consider mean **SigMean** and variance **SigVar** of the slice luminance. We also consider the slice type **Slice_type**, such as IDR or P or B slice, and it is treated as a categorical factor in our model development framework discussed in Section III.

B. Error Frame Factors

We characterize the slice loss in the error frame $e(t) = \hat{f}(t) - \tilde{f}(t)$, by its *amplitude* and *support* (e.g., size, spatial extent, and temporal duration). The size is controlled by slice size either in bytes or number of macroblocks contained in it. The spatial extent is influenced by the number of slice groups and FMO setting in H.264/AVC. The amplitude depends heavily on the underlying video content and the decoder concealment strategy, and may decrease as we progress towards the end of the GOP due to the motion-compensation process [5].

- **Temporal Duration (TMDR)**: It is defined as the temporal error propagation length due to a slice loss. A slice error in a non-reference B frame has a TMDR of 1 since it is not used for predicting other slices, while an error in a reference IDR slice propagates to the end of GOP.
- **Initial Mean Squared Error (IMSE)**: The IMSE of the loss of a slice in a frame is computed between the compressed frame $\hat{f}(t)$ and the reconstructed frame $\tilde{f}(t)$ within the encoder instead of the original uncompressed frame.
- **Initial Structural Similarity Index (ISSIM)**: It is a measure of the structural similarity [6] between two frames.
- **Cumulative Mean Squared Error (CMSE)**: We use CMSE as the ground truth in our model as it is an effective measure of the distortion contributed by a slice loss which also captures the error propagation within the GOP.

III. Model Development

We generated a video database with sequences that have a wide variety of scenes such as a bird's eye view of a city, crowded areas, portraits and still water. These videos were compressed using JM 14.2 reference software of H.264/AVC [8]. The GOP structure was IDR B P ... B with GOP length of 20 frames. The frames were encoded using dispersed FMO and a fixed slice configuration mode where the size of the slice in bytes is predetermined by the user. At the decoder, Motion Copy Error Concealment (MCEC) was used to conceal any slice losses in P and B frames, and spatial interpolation was used to conceal losses in the IDR frames. A training and test set for our model development was formed by randomly splitting the database into a 70:30 ratio, where we train our model on 70% of the data and test on the remaining 30% of the data.

A. Overview of Model Development

We use a generalized linear model (GLM) to predict the CMSE contributed by a single slice loss. Let $\mathbf{Y} = [y_1, y_2, ..., y_N]$ be a vector of our response variable, i.e., measured CMSE values. Each data point in $\mathbf{Y}$ is expressed as a linear combination of a known covariate vector $\mathbf{X} = [x_1, x_2, ..., x_p]$ and a vector of unknown regression coefficients $\beta = [\beta_0, \beta_1, ..., \beta_p]^T$. The regression coefficients are estimated through an Iteratively Re-weighted Least Squares (IRLS) technique. After estimating β, we use it to derive the predicted response variable (i.e, predicted CMSE) vector $\hat{\mathbf{Y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_N]$ computed as $\hat{\mathbf{Y}} = \mathbb{E}(\mathbf{Y}) = g^{-1}(\mathbf{X}\beta)$ where $g(\cdot)$ is a link function.

B. Model Fittng

In the model fitting, a subset of covariates are chosen for the best fit. We use the statistical software R [9] for our model fitting and analysis. The steps for selecting covariates are as follows:

- **Evaluating the Distribution of the Response Variable**: A visual analysis of the measured CMSE distribution revealed that low CMSE values occurred with higher frequency than higher CMSE values. Hence for our model, we classified our response variable as a member of the exponential family of distributions with identity as its link function.
- **Akaike's Information Criterion (AIC)**: We use the AIC index [10] to determine the order in which the covariates are fitted. It is defined as $-2max(L) + 2p$, where p is the number of covariates and L is the log-likelihood estimate for the model.
- **Choosing Covariates**: We let $\mathbf{Y}^k$ represent the model with a subset of k covariates. The i^{th} data point in $\mathbf{Y}^k$, y_i^k, where $i = 1, 2, ..., N$ is expressed as:

  $$y_i^k \quad = \quad \beta_0^k + \beta_1^k x_{i1} + \beta_2^k x_{i2} + \ldots + \beta_k^k x_{ik} + \epsilon_i \qquad (1)$$

  Here, β_0^k is the intercept, β_j^k, $j = 1, 2, ..., k$ are the fitted coefficients, x_{ij} represents the j^{th} covariate for the i^{th} observation in $\mathbf{Y}^k$, and ϵ_i is the error coefficient. The simplest model is the Null Model having only the intercept β_0^k whereas the Full Model has all the p covariates, i.e., $k = p$. We use a forward stepwise approach to choose the covariates.

  Step 1: We fit a group of p univariate models and compute their AIC values. The best univariate model has the smallest AIC value.

  Step 2: We then fit $(p-1)$ multivariate models where each model has two covariates. The first covariate is from the best univariate model in Step 1 and the second covariate is chosen from the remaining $(p-1)$ available covariates. We compute the AIC values for the $(p-1)$ multivariate models and choose the best multivariate model with the smallest AIC value. The two covariates fitted at this stage would progress to the next step to be fitted with the third covariate. This process of fitting covariates is repeated until the stopping criterion is satisfied.
- **Stopping Criterion**: If the model with $k+1$ covariates, has a higher AIC index than the corresponding model with k covariates the process stops. It is also possible that the full model was fitted (i.e., $k = p$) and the stopping criterion has not been satisfied, as was observed during our model fitting with the factors described in Section II.
- **Random Forests**: We improve the performance of our model by introducing two new factors which are interactions between the three most important factors. We use a random forest [11], which is a tree structured classifier, to determine the covariate importance over a large number of decision trees. The trees are grown to the full extent (i.e., trees are not pruned) through binary recursive partitioning. Each response variable data point casts a vote for the most important covariate. Finally random forest outputs the most popular covariates.

TABLE I: Final Model Coefficients in Order of Importance

Name of Covariate	Regression Coefficient
IMSE	1.90×10^{-1}
TMDR	-1.30
MAXRSENGY	-9.88×10^{-9}
ISSIM	-1.91×10^{1}
SigMean	-3.03×10^{-1}
SigVar	-2.86×10^{-3}
MOTX	-2.98×10^{-1}
MOTY	-1.15
AVGINTERPARTS	-1.08×10^{1}
Slice_type.f2	1.20×10^{1}
Slice_type.f3	-1.95×10^{1}
IMSE × TMDR	7.54×10^{-1}
IMSE × MAXRSENGY	1.40×10^{-9}
Intercept	9.45×10^{1}

We observed that IMSE, TMDR and MAXRSENGY are the most important covariates as shown in Table I. We introduced interactions between covariates IMSE and TMDR, and IMSE and MAXRSENGY to improve the model. Intuitively, as IMSE increases, CMSE also increases since slices that are harder to conceal result in higher distortion. As TMDR increases, CMSE increases due to error propagation. The regression coefficients of our final model are also reported in Table I. CMSE shows a positive correlation with IMSE, indicating that if a slice is harder to conceal, its propagative effects would also be greater. Our interaction variables also show similar correlation with CMSE.

IV. Slice Priority Assignment And Problem Formulation

In order to validate our model, we analyze and compare the performance of Quartile Based Prioritization (QBP) on measured and predicted slice CMSE values over an AWGN channel. The predicted CMSE values are computed using GLM derived in Section III-A after deriving video factors while encoding whereas measured CMSE values are computed by decoding videos affected by individual pre-encoded slice losses. We divide the slices from each GOP into 4 priorities based on the quartiles, where priority 1 slices have highest predicted/measured CMSE and priority 4 slices have lowest predicted/measured CMSE.

Our objective is to find the optimal Equal Error Protection (EEP) and UEP code rate allocation for the four priorities in the different bitstreams. We formulate the total expected video distortion of our prioritized data as in [12]. Let R_{CH} be the transmission bit rate of the channel in bits per second. The video is encoded at a frame rate of f_s frames per second, and the total outgoing bit budget for a GOP of length L_G is $\frac{R_{CH}L_G}{f_s}$. The RCPC code rates are chosen from a candidate set $\mathbf{R}$ of punctured code rates $\{R_1, R_2, R_3, ..., R_K\}$. The expected video distortion within the GOP is the sum of the prioritized slice loss distortion over the AWGN channel. The expected distortion of the 'j^{th}' slice depends on the measured/predicted CMSE distortion due to its loss, $D_p(j)$, slice error probability for a given channel SNR, slice size $S_p(j)$ in bits, and RCPC code rate r_i for slice priority i selected from the candidate set $\mathbf{R}$. The optimization problem is formulated as:

$$\min_{\mathbf{r}} \left\{ \sum_{i=1}^{4}\sum_{j=1}^{n_i} \left[1-(1-p_b(SNR,r_i))^{\left(\frac{S_p(j)}{r_i}\right)}\right] D_p(j) \right\}$$

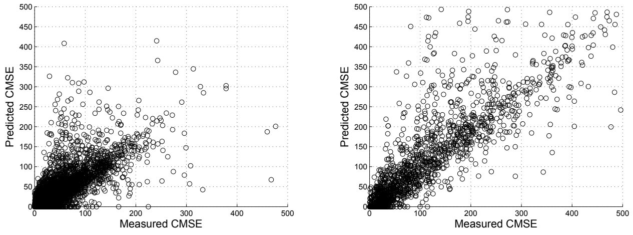
$$\text{subject to}$$

$$\begin{aligned} &(1) \quad \sum_{i=1}^{4}\sum_{j=1}^{n_i} \frac{S_p(j)}{r_i} \le \left(\frac{R_{CH}L_G}{f_s}\right) \\ &(2) \quad r_{i-1} \le r_i \quad \text{for} \quad i = 2,3,4 \end{aligned} \qquad (2)$$

Here n_i is the number of slices of priority i. The formulation only considers slice loss distortion, and ignores compression distortion, and so will be more applicable if compression distortion is negligible compared to slice loss distortion. Constraint 1 in Equation 2 is the channel bit rate constraint and constraint 2 ensures that higher priority slices have code rates which are at least as good as the code rates allocated to the lower priority slices. It speeds up the optimization process by narrowing down the selection set of code rate combinations for the four priorities. The optimization problem is solved using Branch and Bound (BnB) with interval arithmetic analysis [13] to yield the optimal UEP code rates. We also consider EEP based transmission over the AWGN channel where the single strongest code rate that can be used for all the slices within the channel bit rate constraint is determined. Though the final bit rate after adding the parity bits does not exceed the bit budget, there is a possibility that not all of the available bits are utilized due to the set $\mathbf{R}$ being a limited discrete vector of punctured code rates. To be fair, we limit the bit budget of the UEP scheme to the number of bits used by the EEP scheme.

V. Simulation Setup And Experimental Results

We have studied the performance of out CMSE prediction model on CIF Foreman and Tempete video sequences encoded at 512 Kbps and 1024 Kbps using H.264/AVC JM 14.2 reference software [8]. Two different slice sizes, 300 and 900, bytes were used. Other encoding parameters used can be found in Section III. Due to the space constraints, we show results for only Foreman sequence encoded at 1024 Kbps. We use scatter plots to examine the accuracy of our prediction model based on the video factors in Table I. The correlation coefficient, ρ indicates the strength of linearity between the measured CMSE and predicted CMSE values. Figure 1 illustrates that the CMSE prediction model accuracy is high with large correlation coefficient values. It also show outliers, which are data points that were not predicted accurately. These data points result in misclassification of the slices into different priorities.



(a) (b)

Fig. 1: Scatter plot of Predicted CMSE vs. Measured CMSE for Foreman encoded at 1024 Kbps and slice size of (a) 300 bytes (ρ = 0.75), and (b) 900 bytes (ρ = 0.79).

Table II shows the percentage of slices contributed by each frame type in the encoded bitstream. On an average, the IDR, P and B frame slices contribute 25%, 55% and 20% slices, respectively. The share of IDR slices decreases slightly as the slice size increases from 300 to 900 bytes. The converse is true for B slices.

TABLE II: Percentage Distribution of Slices in Foreman

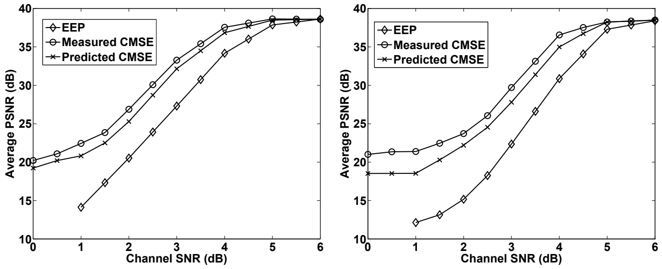
$SliceSize$	IDR	P	B
300	25.9	55.8	18.3
900	22.6	55.6	21.7

Next, we discuss the misclassification of slices in different priorities. If a slice is assigned a priority p_i, such that $i = 1, 2, 3$, or 4, based on the measured CMSE, then we define a **first degree (1°)** misclassification of the slice if it is assigned a priority p_{i+1} or p_{i-1} based on the predicted CMSE. Likewise, a **second degree (2°)** misclassification would result if the slice is assigned a priority of p_{i+2} or p_{i-2} based on its predicted CMSE value. In a **third degree (3°)** misclassification, a slice with the highest priority is assigned the lowest priority or vice versa. A 1° misclassification represents moderate CMSE prediction error and may be tolerable whereas a 2° and 3° misclassifications should be minimized. Also, it is desirable to minimize misclassification of higher priority slices. Table III shows the percentage misclassification for slices from each priority. For the slice size of 300 bytes, less than 20% slices misclassified by 1° belong to priorities p_1 and p_2, and less than 8% and 4% slices are misclassified by a degree of 2 and 3, respectively. For 900 byte slices, less than 15% slices misclassified by 1° belong to priorities p_1 and p_2, and less than 1% of the slices are misclassified by the 2°. The total misclassification is much smaller for 900 byte slices. We observed a similar behavior for Tempete video sequence.

TABLE III: Percentage Slice Misclassification by Degree Corresponding to Each Priority for Foreman using QBP Scheme.

$Degree$	300	900
$1^{\circ}(p_1/p_2/p_3/p_4)$	4.5/14.8/7.9/13.9	5.0/9.4/10.6/7.4
$2^{\circ}(p_1/p_2/p_3/p_4)$	0.9/0.4/4.5/1.9	0.1/0.2/0.1/0.5
$3^{\circ}(p_1/p_2/p_3/p_4)$	0.0/0.0/0.0/3.6	0.0/0.0/0.0/0.0

In addition to slice priority misclassification, we also studied performance proximity of a predicted CMSE based QBP bitstream transmission to that of a measured CMSE based QBP transmission by evaluating the average PSNR (dB) performance of UEP and EEP schemes over an AWGN channel. The mother code of the RCPC code has rate $\frac{1}{4}$ with memory M = 4 and puncturing period P = 8. Log-likelihood ratio (LLR) was used in the Viterbi decoder. The RCPC rates each slice priority can select from were $\{$(8/9), (8/10), (8/12), (8/14), (8/16), (8/18), (8/20), (8/22), (8/24), (8/26), (8/28), (8/30), (8/32)$\}$. The channel bitrate used for our test videos were 2.3 Mbps for the encoding bitrate of 1024 Kbps. For a given video bitrate and slice size, the video quality decreases with decreasing channel SNR due to higher slice error probability caused by more channel errors. Figure 2 shows the average PSNR computed over 100 realizations of each AWGN channel for Foreman encoded at 1024 Kbps with slice sizes of 300 and 900 bytes. Measured CMSE and Predicted CMSE represent the UEP performance for slices prioritized based on measured CMSE and predicted CMSE. At channel SNRs < 1 dB, videos with EEP could not be decoded since too many slices were corrupted. The UEP performance of the predicted CMSE scheme closely follows that of the measured CMSE for both small and large slices even though higher spatial correlation and compression efficiency at larger slice sizes makes it more difficult to conceal. The UEP schemes clearly demonstrate a significant gain over EEP since greater protection is provided to priority 1 slices at the expense of more priority 4 slices being lost over the channel. The gain decreases as the channel gets better.



(a)

(b)

Fig. 2: Average PSNR performance of Foreman video over an AWGN channel. Video was encoded at 1024 Kbps and slice size of (a) 300 bytes, and (b) 900 bytes.

VI. Conclusions

We presented a low complexity scheme to predict the expected CMSE using a generalized linear model. The proposed model used a combination of low complexity parameters which were extracted while the frame was being encoded. Both predicted CMSE and measured CMSE contributions were used to classify slices into different priorities using the QBP scheme. We showed that second degree and third degree priority misclassifications were minimal indicating that we have achieved similar levels of prioritization in our proposed scheme and the UEP performance of the predicted CMSE prioritization over an AWGN channel is close to that of the measured CMSE prioritization.

VII. Acknowledgment

Approved for Public Release; Distribution Unlimited: 88ABW-2012-2761, 11 May 2012. This research was partially supported by awards from the U.S. Air Force Research Laboratory under contract # FA8750-08-1-0078 and FA8750-11-1-0048. Opinions, interpretations and conclusions are those of the authors and are not necessarily endorsed by the United States Government.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[2] A. R. Reibman, V. A. Vaishampayan, and Y. Sermadevi, “Quality monitoring of video over a packet network,” *IEEE Trans. on Multimedia*, vol. 6, no. 2, pp. 327–334, March 2004.

[3] S. Kanumuri, P. Cosman, A. R. Reibman, and V. A. Vaishampayan, “Modeling packet - loss visibility in MPEG - 2 video,” *IEEE Trans. on Multimedia*, vol. 8, no. 2, p. 341, April 2006.

[4] A. Reibman and D. Poole, “Predicting packet-loss visibility using scene characteristics,” in *IEEE Int. Conf. on Packet Video*, November 2007, pp. 308–317.

[5] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman, and A. R. Reibman, “A versatile model for packet loss visibility and its application to packet prioritization,” *IEEE Trans. on Image Processing*, vol. 19, no. 3, pp. 722–735, March 2010.

[6] A. R. Reibman and D. Poole, “Characterizing packet -loss impariments in compressed video,” in *IEEE Int. Conf. on Image Processing*, vol. 5, November 2007, pp. 77–80.

[7] L. Breiman, J. Friedman, C. J. Stone, and R. R. Olshen, *Classification and Regression Trees*, 1st ed. Chapman and Hall, January 1984.

[8] *H.264/AVC reference software JM14.2*, ISO/IEC Std. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[9] The R Project for Statistical Computing. [Online]. Available: http://www.r-project.org/

[10] H. Akaike, “A new look at the statistical model identification,” *IEEE Trans. on Automatic Control*, vol. 19, no. 6, pp. 716–723, January 1974.

[11] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, 2001.

[12] T.-L. Lin and P. C. Cosman, “Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video,” *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 505–508, May 2010.

[13] K. Ichida and Y. Fujii, “An interval arithmetic method for global optimization,” *Springer-Verlag J. of Computing*, vol. 23, pp. 85–97, August 1979.