CCITT SG XV
Working Party XV/1
Experts Group on ATM Video Coding

AVC-310
July 1992

INTERNATIONAL ORGANISATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION

ISO-IEC/JTC1/SC29/WG11

CODING OF MOVING PICTURES AND ASSOCIATED AUDIO

ISO-IEC/JTC1/SC29/WG11
MPEG 92/

**Source:** R. Aravind, C. Horne, A. Reibman, R. Safranek, AT&T
**Title:** Modifications to TM1 to suit Broadcast Applications
**Purpose:** Proposal

## 1. Introduction

The syntax in TM1 is not very friendly towards broadcasting applications. Specifically, the quantization matrices are only allowed to be transmitted at every GOP, by inserting a new sequence header. Hence, in the case of channel tune-in the decoder has to wait until it receives an intra-coded picture (at a new GOP where the quantization matrices are also received) before it can display an acceptable picture. Second, the decoder cannot synchronize its feedback loop to that of the encoder until all macroblocks have been intra-coded once (which is achieved by transmitting an I-picture). These features of the TM require that I-pictures (along with the sequence and GOP headers) be sent frequently in a broadcast situation. This could have an adverse effect on picture quality since I-pictures typically require more bits than non-intra pictures.

## 2. Quantization Matrices

We propose that *for the broadcast profile* all the quantization matrices (and all other high-level information needed for de-quantization) be transmitted in the header of every I- and P-picture. Currently, I-pictures use only one matrix, while two matrices are allowed in P-pictures (for intra and non-intra macroblocks). The overhead for transmitting a matrix is only 64 bytes. Since B-pictures are not themselves used as references, they do not affect channel tune-in and thus do not require the transmission of the quantization matrices; the last transmitted intra and non-intra matrices can be employed for B-pictures at both the encoder and decoder.

### 2.1 Methods for obtaining the matrices

An approach to pick the nonintra quantizer matrix for a P-picture that can improve coding efficiency is to make the matrix coefficient proportional to some function of the variance of the corresponding DCT coefficient for that picture. For a P-picture, an estimate of this variance can be obtained through the encoding of the previous P-picture. Care must be taken to place lower limits on the on the high-frequency matrix coefficients and upper limits on the low-frequency coefficients.

## 3. Leaky Prediction

In P-pictures, "perfect" prediction is currently employed to generate the prediction signal. This type of prediction, while maximizing coding efficiency, has some undesirable side-effects. If the decoder loop gets out of synchronization with the encoder loop, (due to channel errors, for instance) it stays out of sync until the occurrence of the next I-picture. Further, during channel tune-in, the decoder loop cannot achieve synchronization with the encoder loop until the next I-picture.

To alleviate these effects, we propose the use of "leaky" prediction in P-pictures in the broadcast environment. A leaky prediction is obtained by attenuating the perfect prediction pixel-by-pixel (luma and chroma) by a *leak-factor* $LF$, both at the encoder and decoder:

$$\text{leaky\_pred} = (LF^* \, (MCP( \, I_t(x,y) \, ) - 128)) + 128.$$

Here, $I_t(x,y)$ is the input P-picture at time $t$, and $MCP( \, I_t(x,y) \, )$ its (motion-compensated) perfect prediction from the reconstructed picture $I_{t-M}$. The number 128 is chosen so that the decoder loop settles to 128 (mid-grey level) when its input is turned off. It can be observed that the memory in the loop decays rapidly for small values of the leak-factor, and vice-versa.

### 3.1 Implementation of leaky prediction

In this proposal, the leak-factor is restricted to the values of the expression $1 - \dfrac{1}{2^n}$ for integer $n$. The number $n$, which is transmitted for every P-picture, is proposed to be in the range 1-6 (3 bits). The codeword "111" is used to signify perfect prediction (where $n$ is infinite). Since setting $n$ to 0 is equivalent to intra coding, a value of 0 never needs to be transmitted; the codeword "000" is thus disallowed, and an I-picture is sent instead. Also, intra-coded macroblocks in P-pictures are coded as in TM1, without any prediction. In B-pictures, although leaky prediction is not needed for channel tune-in, its addition could make the picture quality more uniform.

Two methods are proposed to implement leaky prediction in the core experiments. The first method is to perform the equivalent multiply first in 16-bit arithmetic followed by the divide with rounding to 8 bits. The complexity of this operation is roughly that of de-quantization; it actually requires less precision than the latter. The second method employs a shift and subtract, exploiting the form of the leak-factor; however, this method uses less precision than the first. Both need to be studied further.

### 3.2 Methods for determining the leak-factor

One method of selecting the leak-factor to use for a particular picture is to treat the leaky predictor as a first-order linear predictor and minimize the variance of the difference between the input picture and the leaky motion-compensated prediction.

Specifically, given the current picture $I_t$, and its motion-compensated prediction $MCP( I_t(x,y) )$ (which requires the previous reconstructed picture $I_{t-M}$ and the motion vectors for $I_t$), then the prediction error with leak is

$$d(x,y) = I_t(x,y) - LF * MCP( I_t(x,y) )$$

and its variance, by taking squares and expectations, is

$$\sigma^2{}_d = ( 1 + LF^2 - 2 * \rho_1 * LF ) \sigma^2{}_{I_t}(x,y).$$

The optimum $LF$ that minimizes this variance is

$$LF_{opt} = \rho_1 = R_{I_t(x,y)MCP( I_t(x,y) )} / \sigma^2{}_{I_t}(x,y)$$

where $R_{I_t(x,y)MCP( I_t(x,y) )}$ is defined as

$$R_{I_t(x,y)MCP( I_t(x,y) )} = \sum_{x=0}^{xmax} \sum_{y=0}^{ymax} I_t(x,y)\ MCP( I_t(x,y) )$$

and $\sigma^2{}_{I_t(x,y)}$, the variance of $I_t(x,y)$, is defined as

$$\sigma^2{}_{I_t(x,y)} = \sum_{x=0}^{xmax} \sum_{y=0}^{ymax} I_t(x,y)\ I_t(x,y).$$

Notice that if $LF_{opt}$ approaches zero, this is a good indication of scene change. When delay is a concern, the correlation $\rho_1$ above can be computed using the last (I, P) or (P, P) pair (not counting the current P). Another option is to use ad-hoc methods to select from the 7 values of $LF$.

A further embellishment would be to allow a different value of $n$ for each color component (Y, Cb and Cr) or (R, G and B) in the picture header. This allows each component to build up and decay with its own time-constant.

### 3.3 Experimental results with fixed leak-factor

Using $M = 3$, we compare an MPEG1 encoder with $N = 12$ to an MPEG1 encoder modified to include a constant leak-factor for P-pictures (the leak is not applied in B-pictures). The leak-factor of $LF$=0.875 ($n = 3$) was chosen so that the two sequences gave equivalent subjective quality. However, Figure 1 shows that the PSNR (luminance) of the sequence with leak is marginally better than the sequence with I-pictures. Both use TM1 rate control at 4Mbps. The images will be shown on D1 tape.

Also shown on tape is the result of a channel change with leak factor 0.875. With $M = 3$, a fair representation of the new image is obtained at the first P-picture, while full image quality is recovered after 21 pictures (7 P-pictures). With $M = 1$, a fair representation of the new image is again obtained immediately, while full image quality is recovered in about 1/4 second.

Figure 1: Flower garden MPEG420