ON THE ENCODING OF THE ANCHOR FRAME IN VIDEO CODING

Lisimachos P. Kondi and Aggelos K. Katsaggelos

Abstract— **In this paper we determine the number of bits to be used for the encoding of the anchor frame in low bit rate video coding in order to improve the quality of the next and subsequent frames to be encoded. This problem is important in a real time video communication system. We use a progressive method for the transmission of the anchor frame. We develop two methods for determining the optimal number of bits to be allocated to the first frame in on line video communication applications.**

I. Introduction

In most approaches known today for low bit rate video coding, no particular attention is given to the effect of the number of bits allocated to the first (intra, anchor) frame of the image sequence on the overall quality of the reconstructed sequence. A fixed Quantization Parameter (QP) or a fixed number of bits is used for the anchor frame. However, this approach is certainly not optimal. Assuming that we are using a constant bit rate channel, the number of bits used for the anchor frame corresponds to a certain time delay. For example, if the bit rate of the channel is R bits per second and we use r bits for the anchor frame, the transmission of that frame will take r/R seconds and we will be able to code the next frame after this time passes. Any frames that become available to the coder while transmitting the anchor frame are discarded. The next frame to be coded will be the one that arrived on or just before the time when the transmission of the anchor frame ended.

Clearly, there should be an optimal time to be alotted to the transmission of the anchor frame. Of course, this time will not be the same for all image sequences. If we spend a lot of bits for the intra frame, we will get a better quality of the reconstructed intra frame. In principle, better quality of the reconstructed intra frame will lead to better quality of the next reconstructed (inter) frame, as long as the time delay is not that great that will cause the correlation of the two frames to be too low. Thus, the time used for the intra frame should be large enough to yield a good quality reconstructed frame but not too large so that the first frame and the next frame to be coded are uncorrelated.

As mentioned earlier, we are primarily interested in low bit rate video coding. Thus, we assume that only one intra frame, the first frame of the sequence, is sent, and all subsequent frames are coded using prediction (inter frames). We based our work on the H.263 video compression standard [1], [2], since it is currently the de facto standard for low bit rate video coding.

To make the decision on when we should stop coding the intra frame and start coding the next (inter) frame we should have some data on the image sequence. Thus, this decision has to be made in real time after the coding of the intra frame has started and some more frames are available to the coder. This leads us to modify the H.263 coder and use a progressive method to code the intra frame, so that we are able to stop coding it at any time. The method we use here is the Embedded Zerotree Wavelet (EZW) method [3], [4], [5].

Here, we assume that our channel is an error-free constant bit rate channel. Our discussion is applicable to this model. There are other channel models that are used for video transmission such as variable bit rate channel models which are suitable for Asynchronous Transfer Mode (ATM) networks, as described in [6].

II. The Embedded Zerotree Wavelet Method

As noted previously, a progressive method is required for the encoding of the anchor frame in order for us to be able to stop the transmission at any time while the decoder is able to decode the anchor frame using the already transmitted bits.

Embedded coding is the same concept as binary finite-precision representations of real numbers. All real numbers can be represented by a string of binary digits (bits). More precision is added to the

L. P. Kondi and A. K. Katsaggelos are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208, USA.

Manuscript received June 13, 1997

representation for each bit we add to the right of the binary number. However, the encoding can stop at any time and provide the best available representation of a real number using the specified number of bits. Similarly, the Embedded Zerotree coder can stop at any time and the bit stream can be decoded.

The EZT method performs a wavelet transform on the image. The wavelet transform coefficients are then encoded progressively. More information can be found in [3], [4].

III. Formulation of the Problem

As mentioned in the introduction, we want to find the best point in time in which we should stop encoding the anchor frame. In the following, we introduce the notation which we will use in the remainder of this paper.

We assume that we want to code an image sequence with a given frame rate in real time. The sequence is to be transmitted over a constant bit rate channel. Let r be the number of bits we use for the first frame, SNR1 the peak signal-to-noise ratio (PSNR) of the reconstructed first frame, and SNR2 the PSNR of the next reconstructed frame. The peak SNR of a 8-bit reconstructed image $\hat{x}(i,j)$ is given by

$$PSNR = 10 \log \frac{255^2}{FD}, \tag{1}$$

where

$$FD = \frac{1}{N \cdot M} \sum_{i=1}^{M} \sum_{j=1}^{N} [x(i,j) - \hat{x}(i,j)]^2, \tag{2}$$

M and N are the dimensions of the image and $x(i,j)$ is the original image.

Our goal is to maximize SNR2 provided that the time delay is acceptable and about the same number of bits is used for the next (second encoded) frame at all times to have a fair comparison.

After the second encoded frame, we use our rate controller to achieve a constant bit rate. We expect the time distance and subsequently the correlation between the rest of the frames to be approximately the same. Thus, we expect that good quality of the second encoded frame will carry over to the next frames, at least in the short run. For this reason we chose the maximization of SNR2 as our objective.

The solution we propose is to treat SNR1 and SNR2 as time series with index the number of source frames that correspond to the bits alotted to the first frame. For example, if the frame rate is 30 frames per second and we give half a second for the coding of the first frame, then, the next frame to be coded will be frame 15. Thus, the PSNR of the first and next frames will be denoted as SNR1[15] and SNR2[15].

We assume that SNR2[n] will have one maximum or, at least, it has one maximum we are interested in, due to our other restrictions, such as acceptable time delay. In the following, we consider two different solutions to the problem, depending on what data we assume we have available to determine the best time when we should stop encoding the anchor frame.

IV. Case 1: Past Values of SNR1[n] and SNR2[n] are Known

Here, we assume that we know the values of SNR1[n] and SNR2[n] up to the corresponding index n (the index which corresponds to the point we are in time). Also, as noted previously, we assume that SNR2[n] has one maximum. Thus, our problem reduces to finding the maximum of SNR2[n]. Experimental results show that SNR2[n] is virtually always monotonically increasing up to its maximum. Thus, we can wait until SNR2[n] starts to decrease and stop at that point. Clearly, we will not get the maximum SNR2[n], since it will have decreased a little. The index n we get using this method is the "optimal" n plus 1. In practice, we do not lose much quality in SNR2[n], since the decline in its value is small.

The drawback of this method is that we need to calculate the value of SNR2[n] for every n in real time. This means that we need to perform the motion compensation, block transform and reconstruction of the second frame for each value of n.

V. Case 2: SNR2[n] is not Known

Let us suppose that we cannot calculate SNR2[n] in real time, but we have instead SNR1[n]. Evaluation of SNR1[n] requires far less computation, thus it is realistic to assume that we have SNR1[n] but not SNR2[n]. However, it is clear that we need a second piece of data in order to estimate the index n which results in the maximum SNR2[n].

As mentioned earlier, SNR2[n] is a function of SNR1[n] and the correlation between the first frame and frame n. A quantitative measure for the correlation can be the frame difference (FD) or better,

the displaced frame difference (DFD). The FD between images $x(i,j)$ and $y(i,j)$ is defined as

$$FD = \frac{1}{N \cdot M} \sum_{i=1}^{M} \sum_{j=1}^{N} [x(i,j) - y(i,j)]^2, \quad (3)$$

while, the DFD between images $x(i,j)$ and $y(i,j)$ is defined as

$$DFD = \frac{1}{N \cdot M} \sum_{i=1}^{M} \sum_{j=1}^{N} [\tilde{x}(i,j) - y(i,j)]^2, \quad (4)$$

where $\tilde{x}(i,j)$ is the displaced original image, after motion compensation is performed.

In order for FD and DFD to be in the same units (dB) as SNR1[n] and SNR2[n], we use the following quantities.

$$FD_{dB} = 10 \log \frac{255^2}{FD}, \quad (5)$$

and

$$DFD_{dB} = 10 \log \frac{255^2}{DFD}. \quad (6)$$

We can assume that we have a system with SNR1[n] and the correlation as inputs and SNR2[n] as output. This system is likely to be nonlinear but, intuitively, we expect it to be memoryless. Thus, the problem reduces to identifying this nonlinear and memoryless system.

Assuming that we use the DFD as a measure of correlation, this is equivalent to estimating the function:

$$SNR2[n] = f(SNR1[n], DFD[n]). \quad (7)$$

The approach we follow is to find some data points of this function using experimental data and then use interpolation to estimate the value of SNR2[n] given SNR1[n] and DFD[n]. The interpolation method we use is the Biharmonic Spline Interpolation [7]. This method allows us to have SNR1[n] and DFD[n] values at "random" points rather than regularly spaced points. In the following, we describe the Biharmonic Spline Interpolation in one and two or more dimensions.

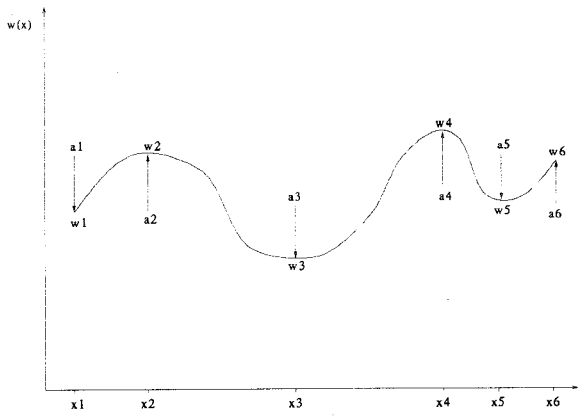


Fig. 1. The biharmonic function $w(x)$ is found by applying point forces to a thin elastic beam or spline

A. *Biharmonic Spline Interpolation in One Dimension*

We wish to find a biharmonic function which passes through N data points. Draftsmen in the 19th century solved the problem by attaching weights to an elastic beam or spline and positioning the weights so that the spline passed through the data points. The forces imposed on the spline by each weight kept it bent. For small displacements, the spline has zero fourth derivative except at the weights (See figure 1). The point force Green function $\phi(x)$ for the spline satisfies the biharmonic equation:

$$\frac{d^4\phi}{dx^4} = 6\delta(x), \quad (8)$$

where $\delta(x)$ denotes the delta function.
The particular solution to (8) is given by

$$\phi(x) = |x|^3. \quad (9)$$

When the Green function $w(x)$ is used to interpolate N data points, w_i, located at x_i, the problem becomes the solution of

$$\begin{aligned} \frac{d^4 w}{dx^4} &= \sum_{j=1}^{N} 6\alpha_j \delta(x - x_j), \\ \text{with } w(x_i) &= w_i. \end{aligned} \quad (10)$$

The particular solution to (10) is a linear combination of point force Green functions centered at each data point, given by

$$w(x) = \sum_{j=1}^{N} \alpha_j |x - x_j|^3. \quad (11)$$

The strength of each point force, α_j, is found by solving the following linear system of equations

$$w_i = \sum_{j=1}^{N} \alpha_j |x_i - x_j|^3. \quad (12)$$

Once the α_j's are determined, the biharmonic function $w(x)$ can be evaluated at any point using (11).

B. Biharmonic Spline Interpolation in Two or More Dimensions

The derivation of the technique in two or more dimensions is similar to the derivation in one dimension. For N data points in m dimensions, the problem becomes

$$\begin{aligned} \nabla^4 w(\mathbf{x}) &= \sum_{j=1}^{N} \alpha_j \delta(\mathbf{x} - \mathbf{x}_j), \\ \text{with } w(\mathbf{x}_i) &= w_i, \end{aligned} \quad (13)$$

where ∇^4 is the biharmonic operator and $\mathbf{x}$ is a position in the m-dimensional space. The general solution is given by

$$w(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \phi_m(\mathbf{x} - \mathbf{x}_j). \quad (14)$$

Again, we find the α_j's by solving the linear system of equations

$$w_i = \sum_{j=1}^{N} \alpha_j \phi_m(\mathbf{x}_i - \mathbf{x}_j). \quad (15)$$

Once the α_j's are determined, the biharmonic function $w(\mathbf{x})$ can be evaluated at any point using (14).

The biharmonic Green functions, ϕ_m for each dimension, are given in Table I. As seen from the table, in two dimensions, as we are interested here, the Green function is equal to

$$\phi_2(\mathbf{x}) = |\mathbf{x}|^2 (\ln |\mathbf{x}| - 1). \quad (16)$$

Num. of Dimensions m	Green Function $\phi_m(\mathbf{x})$
1	$\|\mathbf{x}\|^3$
2	$\|\mathbf{x}\|^2(\ln \|\mathbf{x}\| - 1)$
3	$\|\mathbf{x}\|$
4	$\ln \|\mathbf{x}\|$
5	$\|\mathbf{x}\|^{-1}$
6	$\|\mathbf{x}\|^{-2}$
m	$\|\mathbf{x}\|^{4-m}$

TABLE I
BIHARMONIC GREEN FUNCTIONS

C. Experimental Results

In the following example, we estimate the values of SNR2[n] using actual values of SNR1[n] and DFD_{dB}, using the "Mother and Daughter" sequence. Other sequences were also used and the results were similar when the amount of motion was comparable to that of the "Mother and Daughter" sequence. It should be emphasized that we are interested only in the local maximum of SNR2[n] and not its actual values. In practice, sometimes the "DC level" of the estimated SNR2[n] is different from the original, but in most cases, the index n which corresponds to the local maximum is estimated with very good accuracy. The first few samples of SNR1[n] and SNR2[n] are inaccurate since the EZW method cannot yield acceptable quality images with extremely low bit rates. Thus, in the following simulation, the first five samples of all sequences have been removed.

Figure 2 shows the actual values of SNR2[n] and Figure 3 shows the estimated values of SNR2[n]. The SNR2[n] is computed from the "Mother and Daughter" sequence starting at frame 0 (the first five values are not used). As can be seen, the actual maximum is at index 8, whereas the maximum of the estimated SNR2[n] is at 5. Figures 4 and 5 show actual and predicted SNR2[n] when we start from frame 50 of the "Mother and Daughter'' sequence. In this case, the actual maximum is at index 13 and the estimated maximum is at 9.

Clearly, from these examples we see that the first maximum of SNR2[n] can be estimated quite accurately in most cases using the proposed method.

VI. CONCLUSIONS

The motivation for this work was the intuitive expectation that there should be an "optimal" allo-

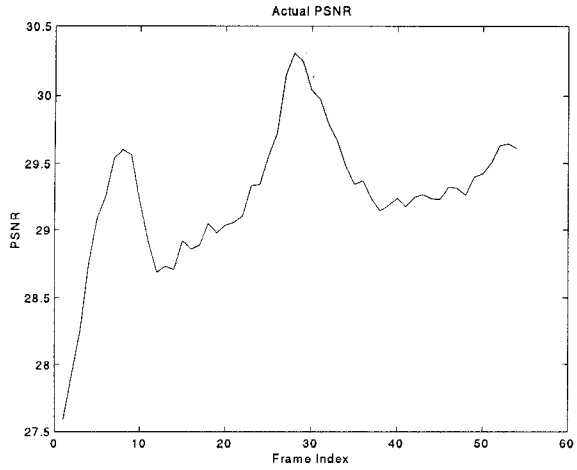


Fig. 2. Actual values of SNR2[n] (in dB). Frame index 0 corresponds to frame 5 of the "Mother and Daughter" sequence.

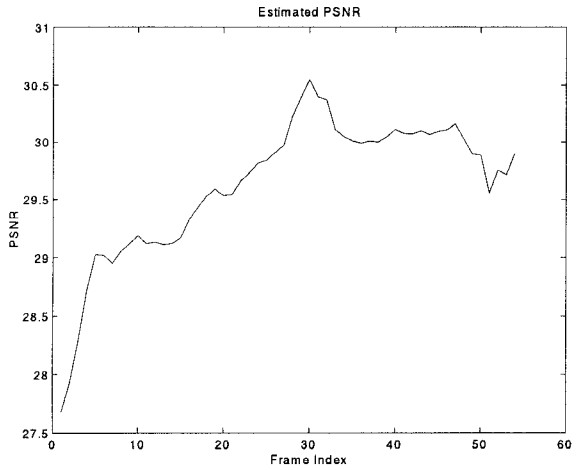


Fig. 3. Estimated values of SNR2[n] (in dB). Frame index 0 corresponds to frame 5 of the "Mother and Daughter" sequence.

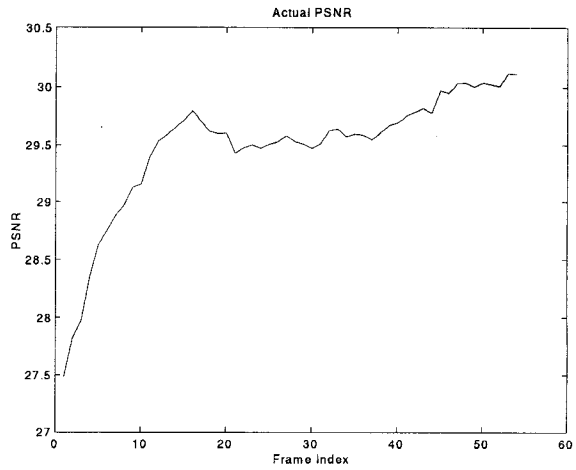


Fig. 4. Actual values of SNR2[n] (in dB). Frame index 0 corresponds to frame 55 of the "Mother and Daughter" sequence.

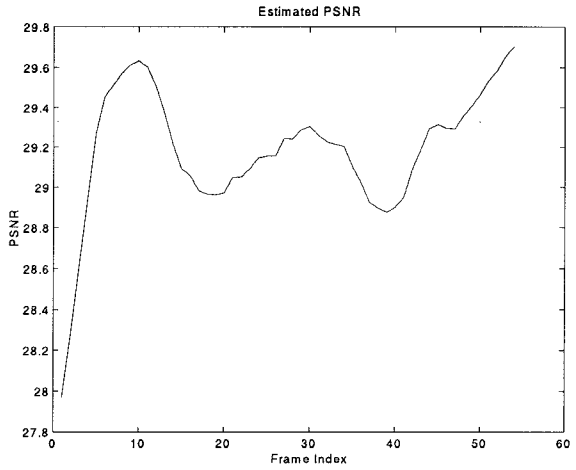


Fig. 5. Estimated values of SNR2[n] (in dB). Frame index 0 corresponds to frame 55 of the "Mother and Daughter" sequence.

cation of bits to the anchor frame which would give us the best quality of the subsequent frame. This bit allocation would depend on the image sequence and should be determined on-line, thus a progressive method for the transmission of the anchor frame should be used.

We used the maximization of SNR2 as our objective since we want to achieve a constant frame rate after the second frame and it is reasonable to assume that the "correlation" between any two subsequent encoded frames will be approximately the same. This will be true unless there is dramatic change in the picture, which is something that cannot be predicted. Thus, better quality of the second encoded frame will usually mean better quality of the rest of the frames, at least in the short run or until there is a dramatic change in the scene.

Experimental results showed that except in cases where there is very little motion, there is a well defined first maximum in SNR2. Thus, it is clear that if we allocate the number of bits for the anchor frame that is specified by the peak in SNR2, we can expect better overall quality of the image sequence, according to the previous discussion.

If computational complexity is no object, it is easy to find the first maximum in SNR2. From ex-

perimental results, we know that SNR2[n] is virtually always monotonically increasing up to the first maximum. Thus, if we can calculate SNR2[n] on-line, we can locate the maximum by pinpointing the location where SNR2[n] starts to decrease. However, the computational complexity in this case is very high since we need to encode and reconstruct the second frame every 1/30th of a second in the case of source frame rate of 30 frames per second.

Future work in this area can involve development of a more sophisticated algorithm for the location of the maximum which will be able to identify a very small local maximum where SNR2 declines a little and then it immediately starts to increase again up to a well defined maximum. The new algorithm should be able to disregard such a small local maximum.

If we assume that we cannot calculate SNR2[n] on-line, we can assume that SNR2[n] is a function of SNR1[n] and the correlation between the two frames, as measured by the energy of the DFD. In this work, we estimate this function using interpolation from experimental data. The computational complexity in this case is much lower, since we do have to perform motion compensation to find the DFD, but we do not have to perform all the other steps that are required in order to find the reconstructed second frame and calculate its PSNR.

The experimental results have shown that using the above method, we can satisfactorily estimate the maximum in SNR2[n]. In most cases, the estimated location of the maximum is very close to the actual location.

Thus, in this paper, we found that it is worthwhile to send the anchor frame progressively and employ methods for determining the best stopping point. The methods described here proved to estimate quite accurately this stopping point.

References

[1] "Video coding for low bitrate communications," Tech. Rep., Draft ITU-T Recommendation H.263, May 1996.

[2] "Video codec test model, TMN5," Tech. Rep., Telenor Research, Jan. 1995.

[3] Jerome M. Shapiro, "An embedded wavelet hierarchical image coder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, Mar. 1992.

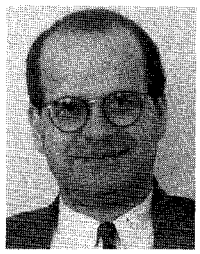
[4] Jerome M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[5] Lisimachos P. Kondi, "On the encoding of the anchor frame in video coding," M.S. thesis, Northwestern University, 1996.

[6] Amy R. Reibman and Barry G. Haskell, "Constraints on variable bit-rate video for atm networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 4, pp. 361–372, Dec. 1992.

[7] David T. Sandwell, "Biharmonic spline interpolation of GEOS-3 and seasat altimeter data," *Geophysical Research Letters*, , no. 2, pp. 139–142, 1987.

Lisimachos P. Kondi was born in Athens, Greece on June 23, 1971. He received the Diploma degree from the Aristotle University of Thessaloniki, Greece, in 1994 and the M.S. degree from Northwestern University, Evanston, IL, USA, in 1996, both in electrical engineering. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering at Northwestern University. His research interests include digital image processing and video compression.



Aggelos K. Katsaggelos received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, in 1979 and the M.S. and Ph.D. degrees both in electrical engineering from the Georgia Institute of Technology, Atlanta, Georgia, in 1981 and 1985, respectively. In 1985 he joined the Department of Electrical Engineering and Computer Science at Northwestern University, Evanston, IL, where he is currently professor. During the 1986-1987 academic year he was an assistant professor at Polytechnic University, Department of Electrical Engineering and Computer Science, Brooklyn, NY. His current research interests include image recovery, processing of moving images (motion estimation, enhancement, very low bit rate compression), computational vision, and multimedia signal processing. Dr. Katsaggelos is an Ameritech Fellow and a member of the Associate Staff, Department of Medicine, at Evanston Hospital. He is a senior member of IEEE, and also a member of SPIE, the Steering Committees of the *IEEE Transactions on Medical Imaging* and the *IEEE Transactions on Image Processing*, the IEEE Technical Committees on Visual Signal Processing and Communications, Image and Multi-Dimensional Signal Processing, and Multimedia Signal Processing, the Technical Chamber of Commerce of Greece and Sigma Xi. He has served as an Associate editor for the *IEEE Transcations on Signal Processing* (1990-1992), an area editor for the journal *Graphical Models and Image Processing* (1992-1995), and he is currently the editor-in-chief of the *IEEE Signal Processing Magazine*. He is the editor of *Digital Image Restoration* (Springer-Verlag, Heidelberg, 1991), and co-author of *Rate-Distortion Based Video Compression* (Kluwer Academic Publishers, 1997). He has served as the General Chairman of the 1994 Visual Communications and Image Processing Conference (Chicago, IL), and he will serve as the technical program co-chair of the 1998 IEEE International Conference on Image Processing (Chicago, IL).