Digital HDTV Compression Using Parallel Motion-Compensated Transform Coders

Hsueh-Ming Hang, *Senior Member, IEEE,* Riccardo Leonardi, *Member, IEEE,* Barry G. Haskell, *Fellow, IEEE,* Robert L. Schmidt, Hemant Bheda, and Joseph H. Othmer

***Abstract*—High Definition Television (HDTV) promises to offer wide-screen, much better quality pictures as compared to today's television. However, without compression a digital HDTV channel may cost up to one Gbits/s transmission bandwidth. We suggest a parallel processing structure using the proposed international standard for visual telephony (CCITT P × 64 kbs standard) as processing elements, to compress the digital HDTV pictures. The basic idea is to partition an HDTV picture, in space or in frequency, into smaller sub-pictures and then compress each sub-picture using a CCITT P × 64 kbs coder. This seems to be, by today's technology, a cost-effective solution to the HDTV hardware. Since each sub-picture is processed by an independent coder, without coordination these coded sub-pictures may have unequal picture quality. To maintain a uniform quality HDTV picture, the following two issues are studied: 1) sub-channel control strategy (bits allocated to each sub-picture), and 2) quantization and buffer control strategy for individual sub-picture coder. Algorithms to resolve the above problems and their computer simulations are presented.**

I. Introduction

In the North American digital hierarchy, 44 Megabits/s is approximately the rate of the fairly ubiquitous DS3 transmission facilities, as well as the upcoming ISDN H22 rate. Thus, it is an interesting and potentially useful rate to consider for the transmission of High Definition Television (HDTV).

One of the main difficulties for near future coding of HDTV is the relatively high speeds required for the digital processing circuitry. This problem is exacerbated if state-of-the-art compression algorithms are contemplated.

This complexity coupled with the high speeds required can significantly impact the costs of the codecs. Fast static random access memories (SRAMS) may have to be used instead of the slower and cheaper dynamic random access memories (DRAMS). Gallium arsenide integrated circuits may have to be used instead of silicon. Custom IC's will be needed instead of the currently cheaper general purpose digital signal processors (DSP's).

A possible solution to this dilemma is to use parallel processing of slower and therefore cheaper circuits to accomplish the overall task. In a pipelined system, each processor would do a portion of the job and pass the result to the next processor. However, with this arrangement each processor must still handle all the pels at the high speed of the original picture.

We believe a better solution is to divide the HDTV picture into sub-pictures and use slower processors to handle each sub-picture. With this approach, high speed input and output of pels occurs only in bursts. Actual processing of pels by each processor can take place at a much slower rate between the bursts of i/o.

A very significant benefit accrues if we choose as the basic processor a standard ISDN videophone codec. Not only is the cost per processor lower because of the lower speed, it will also be cheaper because of economies of scale. With a growing market for ISDN videophone and video conferencing, the costs of a basic codec will soon plunge dramatically.

A problem that must be solved with such an arrangement is how to allocate bit-rate amongst the various codecs. Assigning the same rate to each is inefficient and may produce different image quality in different parts of the picture. Several solutions are discussed and evaluated.

In this paper we investigate two possibilities for subdividing the picture. First we look at pel domain subdivision of a Zenith [1] format HDTV picture picture into 12 sub-pictures, each of size 352 pels by 240 lines or less. Next we look at frequency domain subdivision into 16 sub-bands, each subsampled to produce a sub-picture as is done above.

II. Standard ISDN Video telephone Codec

The recently negotiated CCITT P × 64 kbs standard [2] for ISDN videotelephone utilizes a number of methods for image data compression. For example, color information is subsampled 2:1 compared with luminance to form common intermediate format (CIF) frames of 352 pels by 288 lines for luminance and 176 pels by 144 lines for the two Cr and Cb color signals.

With conditional replenishment, only the parts of the picture that change from frame to frame are sent.

Using motion compensated predictive coding, a frame-to-frame motion estimate is first calculated for the moving areas of the frame to be sent. Then a prediction of these current-frame pel values is obtained by spatial displacement of pels in the previous frame by an amount corresponding to the aforemeasured motion estimate. The difference between the current frame pels and their predicted values is then coded and sent along with the corresponding motion estimates.

At the receiver, the motion estimates are used to calculate

Manuscript received January 2, 1991; revised March 7, 1991. This paper was presented in part at SPIE Visual Communications and Imaging Processing '90, Lausanne, Switzerland, October 1990. This paper was recommended by Associate Editor Yasuhiko Yasuda.

H.-M. Hang, B. G. Haskell, R. L. Schmidt, H. Bheda and J. Othmer are with AT&T Bell Laboratories, Holmdel, NJ 07733.

R. Leonardi is with the Signal Processing Laboratory of EPFL, Lausanne Switzerland.

IEEE Log Number 9100840.

1051-8215/91/0600-0210$01.00

the same prediction as at the transmitter. The received difference or "prediction error" is then added to this prediction to produce the decoded pels for display.

The prediction errors are not transmitted directly. Instead they are first transformed to the frequency domain via the "discrete cosine transform" (DCT). The resulting transform coefficients are then quantized, and the nonzero values coded for transmission. The decoder must inverse transform the received information in order to obtain correct prediction error values.

The CCITT P $\times$ 64 kbs algorithm gains efficiency in the transform, as well as in addressing the pels to be sent by first dividing the CIF frames into blocks of 8×8 pels each. Four blocks of luminance and one of each color are then clustered to form a macro-block of six blocks. Thirty-three macroblocks form a group-of-blocks (GOB), and 12 GOB's form a complete CIF Y-Cr-Cb frame.

Buffer control is mostly outside the standard, except that no encoder is allowed to overflow or underflow the buffer of a hypothetical reference decoder (HRD). The HRD buffer size is approximately $0.133R + 256000$ bits, where R is the video bit rate (bits/s). For broadcast applications, codec delay is less of an issue, and buffer size could be larger. However, buffer overflow and underflow must be avoided whatever the buffer size.

III. Spatial Decomposition

In this section, we describe a spatial decomposition approach that partitions an HDTV sequence in the spatial domain into a number of sub-sequences. Each subsequence is processed by a P $\times$ 64 kbs coder separately. Our target total channel rate is 44 Mbps. Although the Zenith proposed HDTV format is used as an example in the following discussions, the concepts of our algorithms can easily be extended to the other picture formats.

The spatial domain decomposition is shown in Fig. 1. To reduce the inter-coder communications and to simplify the hardware complexity, we assume no direct information exchange among these P $\times$ 64 kbs coders. When necessary, each of these coders is called a "subcoder" in this paper to distinguish them from the entire HDTV coder (consisting of 12 or 16 sub-coders). A Zenith proposed HDTV picture, 720 lines by 1360 pels, is about 2.5 times vertically and four times horizontally the size of the common intermediate format (CIF, 288 lines by 352 pels) defined in the P $\times$ 64 kbs standard. In our experiments, twelve P $\times$ 64 kbs codecs are used. Their picture sizes are described in Fig. 1. The Zenith format adopts the progressive pel scan order, which agrees with the CIF. However, its temporal resolution, 60 frames/s, is twice that of the CIF (30 frames/s). Hence, we have to assume the P $\times$ 64 kbs codecs we use can work at 60 Hz frame rate.

One of the concerns is the motion range between two successive HDTV frames. The P $\times$ 64 kbs standard can handle displacements up to $+/-$ 15 pixels. As the temporal resolution is doubled (60 Hz), it implies that our codec can track motion to a maximum displacement of $+/-$ 900 pels/s.

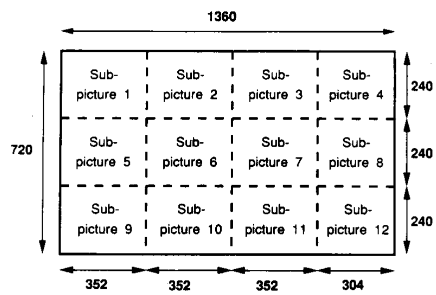


Fig. 1. Spatial decomposition of an HDTV frame.

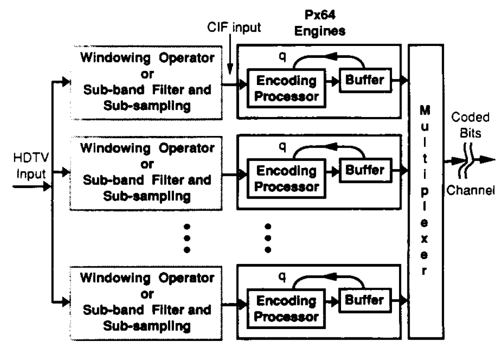


Fig. 2. Functional diagram of parallel HDTV encoder.

This is roughly equivalent to 1.5 s for an object to move horizontally across the entire picture, a very fast movement.

The primary focus of our study is a good channel allocation strategy for distributing the available bandwidth to each processor, and a good buffer feedback control strategy for adjusting the quantizer in each sub-coder. Adjusting the quantizer step-size frequently allows the output buffer to be small. It also allows the encoder to track the local variations inside a picture. However, this can result in a nonuniform quality in the reconstructed picture due to the different step sizes used in quantizing various parts of a picture. An uneven picture quality may also be caused by an inappropriate channel bandwidth allocation as we will discuss in Section III.-A.

Fig. 2 shows a high-level block diagram of an HDTV coder system that uses either a spatial decomposition or a sub-band decomposition. Detailed hardware implementation of such a system is not displayed here. Conceptually, the individual buffer inside each sub-coder feeds compressed data to a global buffer through a multiplexer. The constant-bandwidth external channel is connected to this global buffer. Inside a subcoder, the content of its output buffer is normally used to control the quantization step size for that particular sub-coder.

A. Results of Two Simple Coders

As a first attempt, we use a uniform bit-allocation strategy, in which each processor gets assigned the same proportion of the channel bandwidth. Two quantizer adjustment intervals are under investigation. In the first one, the buffer content is examined and the quantization step may be adjusted at an interval of 11 macro-blocks, a row of GOB. In the second one, a more uniform picture quality is sought through a single adjustment of the step size only once per frame.

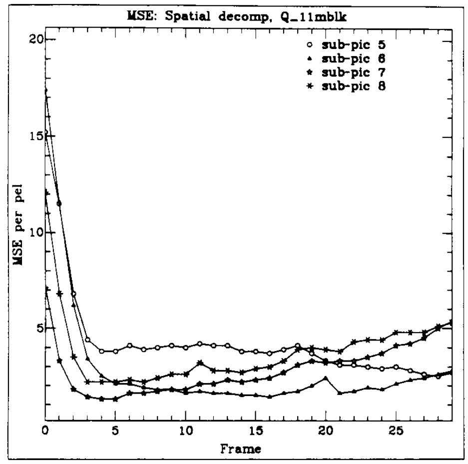
In both cases, their quantization step sizes are adjusted by the same formula—the linear function proposed by Reference Model number 8 (RM8) [3]. That is, the step size used for the next 11 macro-blocks (case 1) or the next frame (case 2) is linearly proportional to the fullness of the buffer content at the moment it is examined. According to the $P \times 64$ kbs standard, the dynamic range of transform coefficients is -2048 to 2047, the minimum quantization step size is 2, and the maximum step size is 64. Only the even values are allowed (32 levels in total).

Fig. 3(a) shows the mean square errors produced by the RM8 encoder for the sub-pictures 5 to 8. Except for the first frame (intra-coded frame), the average mean square errors due to coding are generally very small. These low mean square errors are also reflected on the reconstructed picture quality—coding defects are almost invisible. However, some sub-pictures are harder to compress as compared to the others, and hence they produce higher mean square errors when an equal amount of bits are assigned to every sub-picture.

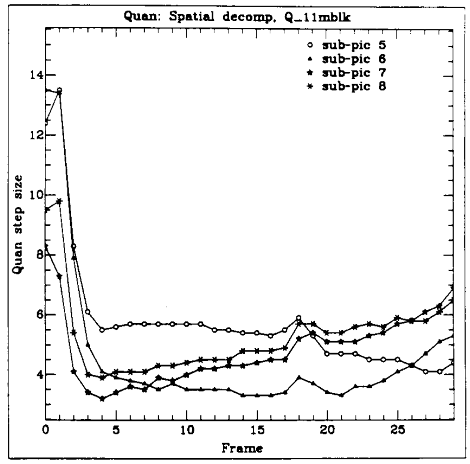
The above results are consistent with the average quantization steps used in these sub-pictures as shown in Fig. 3(b). The difficult sub-pictures, the first 17 frames of sub-picture number 5 for example, have to use a larger quantization step size in order to match the assigned channel bandwidth. Fig. 3(c) shows the bits generated per frame by this coder. Because of the frequent adjustment of quantization step size in RM8, except for the first a few frames, the average bits per frame are about the same for all the sub-pictures and for every frame.

Clearly, the equally distributed channel strategy is less desirable because certain sub-pictures may have lower picture quality due to the use of higher quantization steps. The uneven picture quality across a full HDTV frame can be annoying to the human viewers. This may become a more serious problem when a picture sequence contains several objects moving at different speeds and with varying shapes. Some sub-pictures may contain mostly the stationary background and hardly any new information has to be transmitted, while the others change their contents drastically and thus a large number of bits are needed to update those pictures. A more desirable channel allocation strategy is to distribute the bit rates according to the "complexity" of the sub-picture contents. In our example, sub-picture 5 should get assigned more bits so that its quantization step size can match the others'. Consequently, its mean square errors are reduced and can match the mean square errors of the other sub-pictures. This is the first problem we try to solve, namely, a better channel rate allocation strategy.

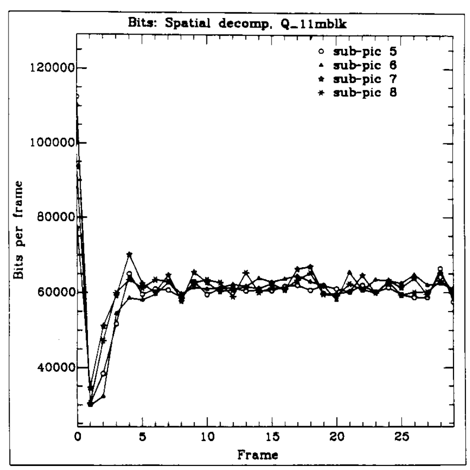
Besides the non-equal quantization levels among sub-coders, there is a second factor that may result in uneven quality coded pictures. In RM8, the quantizer is adjusted at every 11th macro-block. Hence, the step size may vary from one part of a sub-picture to another part. This phenomenon is



(a)



(b)



(c)

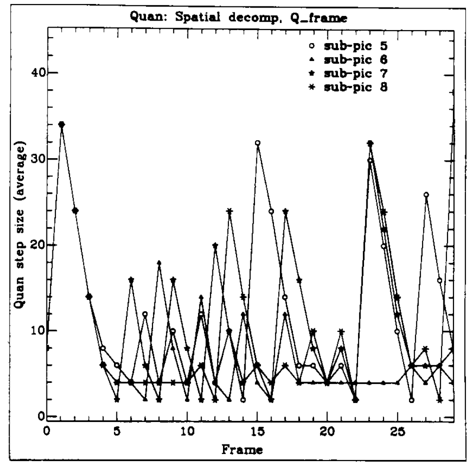
Fig. 3 (a). Mean square errors of spatial decomposition using RM8—quantizer adjusted at every 11th macro-block. (b) Quantization step sizes of spatial decomposition using RM8—quantizer adjusted at every 11th macro-block. (c) Bits per frame of spatial decomposition using RM8—quantizer adjusted at every 11th macro-block.

clearly seen at the beginning of a picture sequence (or a scene change) where a large amount of bits are produced for the intra-coded frame (the first frame, say) and thus the quantization step gets rather large by the end of that frame. In the following frame, the step size is reduced significantly because it often is an inter-coded frame that generates much less bits. The variation of step sizes can, therefore, be very significant within a single frame. A simple proposal to amend this second problem is to freeze the quantization step size inside a sub-picture. In other words, a single step-size is used through out an entire sub-picture.

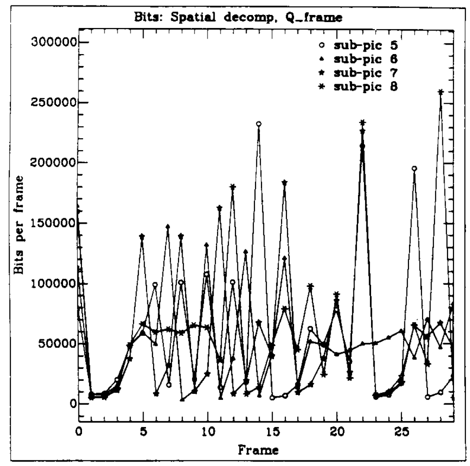
Fig. 4 shows the results of this simple quantization adjustment strategy. A fixed small quantization step size, expected to produce good quality pictures, is chosen for the first intra-coded frame. As expected a very large number of bits are generated. Then we wait for the buffer to clear the bits until it reaches the half-full mark before processing the next frame. For simplicity, in our simulations we assume all the subcoders continue with a half-full buffer at the beginning of the second frame. On the following pictures, the quantizer of each sub-picture is adjusted independently once per frame. Again, every subcoder is assigned the same bandwidth.

It is clear from Fig. 4(a) that the simple RM8 quantization adjustment formula does not perform well when the quantizer is updated only once per frame. In sub-picture 5, for example, the step size oscillates between two extreme values. It starts with a small value and thus a lot of bits are produced. Its output buffer is nearly full; hence, a large step (according to the RM8 linear formula) is chosen to reduce the bits for the next frame. On the next frame, this large step generates a small number of bits, and thus the buffer is nearly empty at the end. Therefore, a small step is again chosen for the following frame and the above pattern repeats. The same pattern can also be observed from Fig. 4(b), the bits produced per frame.

As a result of this undesired quantizer step oscillation, the coded picture quality varies significantly with time. In addition, buffer underflow and overflow occur from time to time due to the sudden decrease and increase of the coded bits. This is the second problem we would like to address, namely, a better quantizer adjustment strategy for the individual subcoder.



(a)



(b)

Fig. 4 (a) Quantization step sizes of spatial decomposition—quantizer adjusted once per frame. (b) Bits per frame of spatial decomposition—quantizer adjusted once per frame.

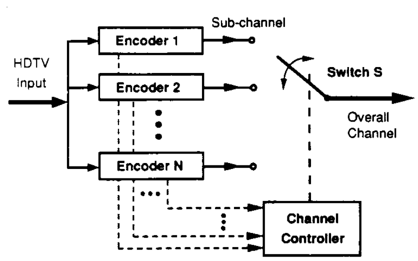


Fig. 5. Block diagram of dynamic channel allocation for a parallel HDTV encoder.

B. A Dynamic Channel Allocation Algorithm

Fig. 5 shows the structure of a video encoding system with N sub-coders. Switch S controls the bit-rate that each sub-coder is actually using. During T units of time, the encoder i is connected to the main channel $f_i \cdot T$ time units, where f_i is a fraction number. Therefore, the sub-channel rate is $C_i = f_i \cdot C$ where C is the overall channel bandwidth, i.e., 44Mbps in our example. The purpose of a channel bit allocation scheme is to ensure that the quality of all sub-pictures is compatible. From a practical application viewpoint, a simple channel allocation procedure with few calculations and little information exchange is preferred. Therefore, we assume the channel assignment is updated only once per frame, and each subcoder only provides two pieces of information to the channel controller: the average quantization step size and the average bits per pel in coding, for every picture frame.

Our goal is to choose a set of *channel sharing factors*, $\{f_i, i = 1,\cdots, N\}$, one for each sub-coder, that will lead to equal coding errors in the sub-pictures. If, for a given distortion, we know the bits used to encode every sub-picture, then the channel sharing factor of a particular sub-picture can be set proportional to its required bits in coding. Therefore, we need a model (a relationship) that relates distortion (d) and bits (b) for coding. This model may be derived either from data empirically or from theory under proper assumptions. We will first derive this model from theory and then verify it with the known data in the literature.

Assume that the coding process in each sub-coder is equivalent to an entropy-coded uniform quantizer operating at low distortion levels. This assumption seems reasonable because of P × 64 kbs coder is essentially a uniform quantizer performed on the DCT coefficients followed by a variable-word-length coder that approximates an ideal entropy coder. Furthermore, the coding errors in this system are expected to be very small since it is designed for HDTV use. If every transform coefficient can be viewed as an i.i.d. (independent identically distributed) signal source, the quantizer parameters are well approximated by the following equations [4]: (for the kth transform coefficient):

$$d(k) = V(k) \cdot e^{-\alpha \cdot b(k)}, \tag{1}$$

and,

$$d(k) = \beta \cdot q^2(k), \tag{2}$$

where $d(k)$ is the mean square quantization error, $b(k)$ is the bits in coding, $q(k)$ is the quantization step size, $V(k)$ is a signal dependent parameter (proportional to the signal variance), and the values of α and β are decided by the signal probability distribution. If the probability distribution of a transform coefficient is either uniform, Gaussian, or Laplacian, then α is 1.39, and β is 1/12 [4].

Assuming that there are in total L transform coefficients to be coded, where L is 64 for the threshold transform coding strategy adopted by the P × 64 kbs standard. Then, we multiply all the (1) of every transform component and obtain

$$d(1)d(2)\cdots d(L) = V(1)V(2)\cdots(L) \cdot e^{-\alpha[b(1)+b(2)+\cdots+b(L)]}. \tag{3}$$

In a P × 64 kbs coder, the same quantization step size is applied to all the transform coefficients in a macro-block;

$$q(1) = q(2) = \cdots = q(L) = q. \tag{4}$$

Hence, from (2),

$$d(1) = \beta \cdot q^2 = d(2) = \cdots = d. \tag{5}$$

Therefore, (3) can be simplified to

$$d^L = V(1)V(2)\cdots V(L) \cdot e^{-\alpha \sum_{k=1}^{L} b(k)}, \tag{6}$$

or

$$d = E \cdot e^{-\alpha b}, \tag{7}$$

where $E = [V(1)V(2)\cdots V(L)]^{1/L}$ is a signal-dependent parameter, $b = \frac{1}{L}\sum_{k=1}^{L} b(k)$, the average bits per sample, and $d = \beta \cdot q^2$, the mean square error per sample.

Strictly speaking, (1) and (2) may not be a very accurate model of a threshold transform coding system with a specific strategy in variable-word-length coding and transform coefficient selection. Particularly, the parameters α, β and L may have to be derived from data if, in fact, (5) and (7) are usable. However, since the video signal before transform coding is often modeled by a first-order Gauss–Markov model, (7) generally represents a reasonable approximation to the rate-distortion function of such a signal source. In addition, our purpose is to estimate the values of certain variables in the immediate future in terms of the same variables in the past. From our experiments, the rough model and parameters we use seem to be adequate for serving that purpose. Also, (5) and (7) are consistent with the pictorial data reported by Netravali and Haskell [5].

Assuming that the above model is valid for every sub-coder, we have

$$d_i = E_i \cdot e^{-\alpha \cdot b_i} = \beta \cdot q_i^2,\ i = 1,\cdots,N, \tag{8}$$

where E_i is a picture-dependent parameter, α and β are two picture-independent parameters (assuming the transform coefficients have either uniform, Gaussian, or Laplacian distributions), q_i represents the average quantizer step size, and b_i represents the bits per pel produced by the sub-coder i.

Let us denote by $E_{i,n}$ the characteristic parameter of sub-picture i at time n, and $q_{i,n}$, $b_{i,n}$ and $d_{i,n}$ are defined similarly. In order to achieve the same d_i for all i, the quantizer step should be the same for every sub-picture; thus we would like to have

$$q_{i,n} = q_n \text{ for all } i. \tag{9}$$

We assume that $E_{i,n} = E_{i,n-1}$, i.e., the characteristic of a sub-sequence does not change very much between two neighboring frames. In addition, we estimate the distortion of every sub-picture at time n by the average distortion of all the sub-pictures at time $n-1$,

$$\hat{d}_{i,n} = \hat{d}_n = \frac{1}{N}\sum_{i=1}^{N} d_{i,n-1} = \frac{1}{N}\sum_{i=1}^{N} \beta \cdot q_{i,n-1}^2. \tag{10}$$

Under the above assumptions, the estimated bits per pel for sub-picture i at frame n, can be expressed by

$$\hat{b}_{i,n} = \frac{\ln\left(E_{i,n-1}/\hat{d}_n\right)}{\alpha}, \tag{11}$$

where $E_{i,n-1}$ is computed from (8)

The channel sharing factors can, therefore, be calculated by

$$f_{i,n} = \frac{\hat{b}_{i,n} \cdot \text{pels\_in\_subpicture\_i}}{\sum_{i=1}^{N}\left(\hat{b}_{i,n} \cdot \text{pels\_in\_subpicture\_i}\right)}. \tag{12}$$

In reality, a sub-coder may not work at channel rates higher than a certain limit; hence, we pose an upper limit on the sharing factor, $f_{\max}$. Picture-dependent parameter E may have a sudden change during a scene change. To insure a minimum channel bandwidth for every sub-coder, a lower limit, $f_{\min}$, may help reduce the buffer overflow problem during scene changes. A minimum sharing factor can also reduce the effects caused by the incorrect model parameters evaluated at very low bit rates. Our experiments seem to indicate that this model and/or our parameters are less accurate at rates lower than 0.2 bits/pel. There are two reasons. First, (1) and (2), in theory, are inaccurate at very low rates. And second, in reality, the other components such as macro-block and group-of-block (GOB) overheads and motion vectors in P × 64 kbs codes are not negligible at very low bit rates, while our model is derived for transform coefficients only.

Finally, we may prefer a smoother change of the channel sharing factors; the previous sharing factors can then be incorporated by the following formula:

$$f_{i,n} = \text{weight} \cdot f_{i,n-1} + (1 - \text{weight}) \cdot f_{i,\text{compute}}, \tag{13}$$

where $f_{i,\text{compute}}$ is the f_i computed from (12).

We first apply this simple algorithm to the RM8 encoder. In our simulations, the lower limit of the channel sharing factors is 0.017, 20% of the initial value (which is 1/12). The upper limit is 0.25, three times the initial value. The *weight* in (13) is chosen to be 0.3.

Fig. 6(a) and (b) are the mean square errors and the quantization step sizes, respectively, produced by employing the above dynamic channel allocation scheme. Each of these quantities from all the sub-picture coders has similar values, as we desire. (The simulation was conducted on all the sub-pictures; to save space only four coders are shown here.) For our test pictures, almost no visible coding artifacts can be detected on the reconstructed pictures. As we expect, the channel rates assigned to each sub-picture is different and is adjusted for every frame as shown by Fig. 6(c). Although we cannot prove mathematically the convergence of this dynamic channel allocation algorithm, it seems to converge well in our simulations.

C. A Quantization Step Assignment Algorithm

Our goal is to adjust the quantization step size only once per frame, while still maintaining a stable and even picture quality over time. With a large amount of measured data and operations, such as pre-analyzing the pictures to be coded, we may have a good estimate of the quantization step size that should be used. However, to save computation and complexity, a simple scheme is proposed, employing the same model suggested in Section III-B. Only two pieces of the previously coded picture information, quantization step size and bits per pel, are used to estimate the quantization step size for the next picture frame.

Assuming that (8) is valid for a P × 64 kbs coder, that is

$$d = \beta \cdot q_{n-1}^2 = E_{n-1} \cdot e^{-\alpha \cdot b_{n-1}}, \tag{14}$$

where b_{n-1} is the average bits per pel, and q, the quantization step size, measured at frame $n-1$. Therefore, E_{n-1} can be obtained by

$$E_{n-1} = \beta \cdot q_{n-1}^2 \cdot e^{\alpha \cdot b_{n-1}}. \tag{15}$$

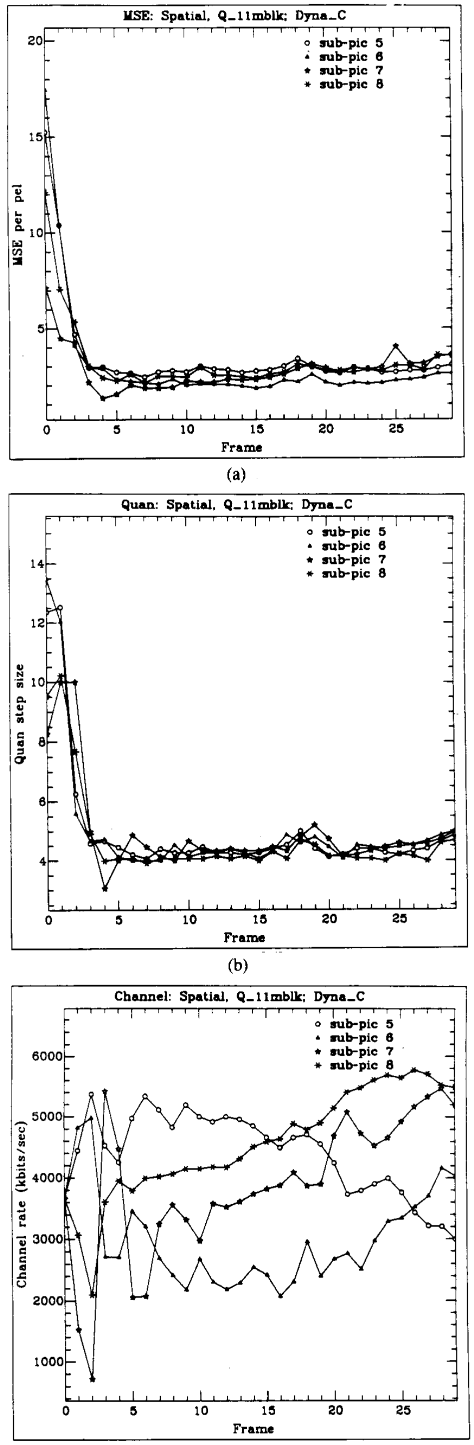


Fig. 6 (a) Mean square errors of dynamic channel allocation algorithm applied to spatial-decomposition coder. (b) Quantization step sizes of dynamic channel allocation algorithm applied to spatial-decomposition decoder. (c) Channel bit rates of the dynamic channel allocation algorithm applied to spatial-decomposition coder.

Next, we calculate the bits expected to be produced at frame n by

$$\hat{b}_n = (b_{\text{desired},n} - B_{n-1} + C \cdot T)/\text{pels\_in\_a\_frame}, \quad (16)$$

where C is the channel rate in bits/s, T is the time for processing frame n and is 1/60 s in our constant frame rate case, B_{n-1} is the measured buffer content at the end of frame $n-1$, and $B_{\text{desired},n}$ is the desired buffer content at the end of frame n, which is chosen as 50% of the maximum buffer size.

If the picture characteristic parameter E is almost constant between two adjacent frames, the new quantization step size can be computed by using (14) again for frame n, i.e.,

$$q_n = \left[\frac{E_{n-1}}{\beta}\right]^{1/2} \cdot e^{-\frac{\alpha}{2}\cdot \hat{b}_n}. \quad (17)$$

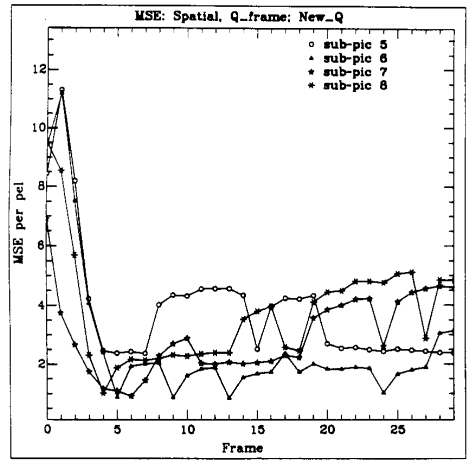
We first apply the above quantizer adjustment scheme to the constant channel coding system described in Section III-A. The results are shown in Fig. 7. The mean square errors, in Fig. 7(a), are now fairly stable and maintain about the same values after the first few frames. So does the quantization step size in Fig. 7(b) and (c). Since the P × 64 kbs quantizer cannot use the values between 4 and 6, it may have to bounce between these two values to produce an average bits that uses, if allowed, a step-size in between. During our simulations, no buffer underflow and overflow appeared.

In addition to this new quantization scheme, if the dynamic channel allocation algorithm proposed in Section III-B is also used, the results are shown in Fig. 8. The mean square errors of all the sub-pictures, Fig. 8(a) are now closer to each other because of the proper assignment of the channel rates. The quantization step sizes are now all around 4 and are fairly stable. Comparing Fig. 6(c) and Fig. 8(c), the channel rate curves in the latter are not as smooth. This may be due to the fact that the quantization step adjusted by the quantizer inside a sub-coder is different from the step size expected by the channel allocation algorithm, since these two values are chosen independently. A more intelligent channel controller which also receives the buffer information from every sub-coder and decides both the quantization step and the channel rate consistently may perform even better.

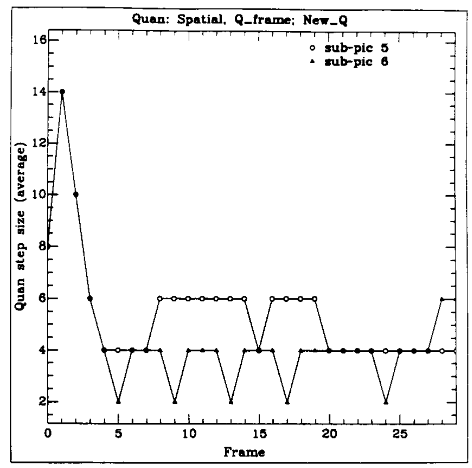
IV. Sub-Band Decomposition

This section describes a sub-band decomposition of an HDTV sequence into 16 bands. Each band is then processed by a P × 64 kbs coder.

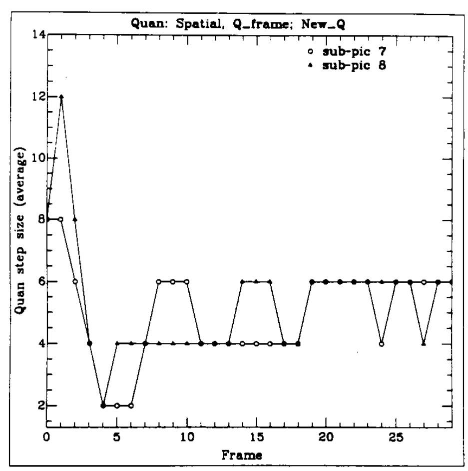
There are advantages and disadvantages associated with the sub-band decomposition as compared to the spatial decomposition. An advantage is that the lowest sub-band has characteristics similar to the CIF size pictures. Hence, the various parameters used in P × 64 kbs such as block size, maximum motion range, and variable-word-length tables may match this lowest band images better. Another advantage would be that certain (higher) bands may contain little information and may be coded with very few bits to save processing and transmission.



(a)



(b)



(c)

Fig. 7 (a) Mean square errors of new quantizer adjustment algorithm—quantizer adjusted once per frame. (b) quantization step sizes of new quantizer adjustment algorithm—quantizer adjusted once per frame. (c) Quantization step sizes of new quantizer adjustment algorithm—quantizer adjusted once per frame.

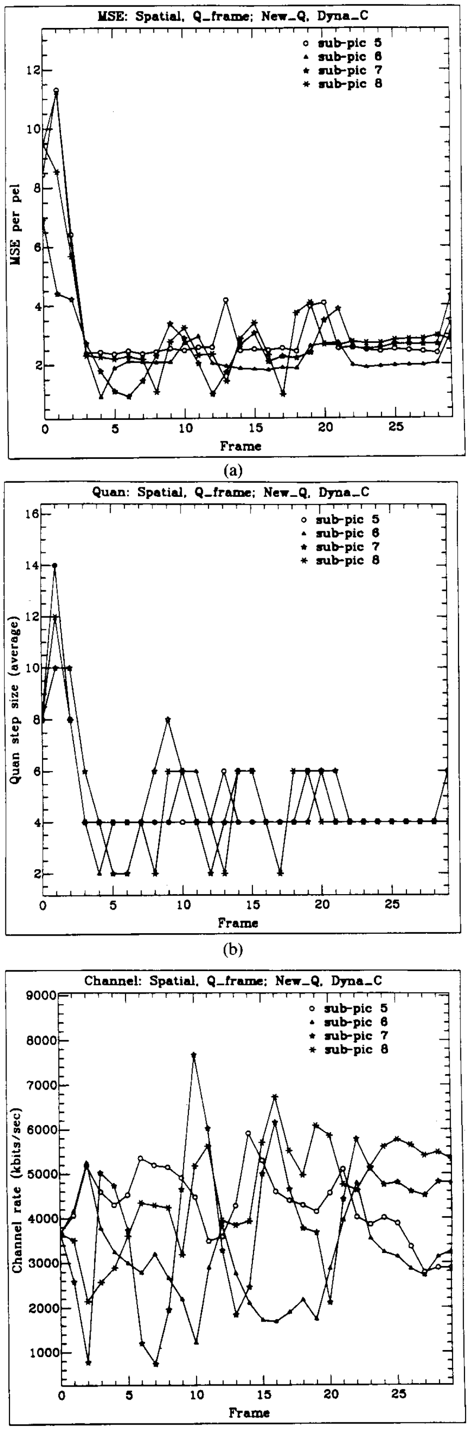


Fig. 8 (a) Mean square errors of using both new quantizer adjustment algorithm and dynamic channel allocation algorithm (quantizer adjusted once per frame). (b) Quantization step sizes of using both new quantizer adjustment algorithm and dynamic channel allocation algorithm. (c) Channel bit rates of using both new quantizer adjustment algorithm and dynamic channel allocation algorithm.

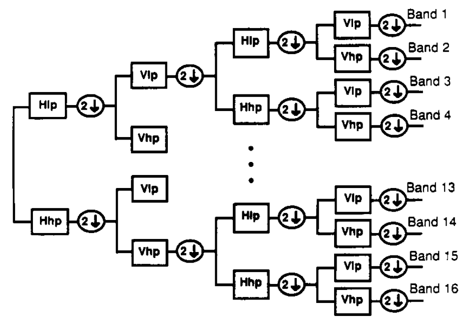


Fig. 9. Functional Diagram of HDTV encoder using sub-band decomposition.

On the other hand, the lowest band processor is expected to operate at a much higher bit rate since it contains most of the full picture information. This may require special hardware implementation for that particular sub-coder. A good channel allocation scheme is even more critical for this coder, because some of the bands are much more important than the others and bit allocation needs to be assigned accordingly. Since the picture content is varying with time, a fixed bit-allocation table is not expected to perform well all of the time.

Furthermore, our simple system structure, which assumes no information exchange among the sub-coders, may not be very efficient in compression. The information among various sub-bands is clearly correlated, for example, motion vectors and edge locations. However, this correlation cannot be exploited unless these sub-coders could share information. Finally, the non-perceivable quantization noise introduced at the sub-bands to adjust the intensity range may become significant and visible after the pictures are synthesized. This problem will be discussed in detail at the end of this section.

Fig. 9 shows the process of splitting an HDTV frame into 16 bands, using a 2-stage filter bank. After each filter stage, signals are sub-sampled by 2. At the end of the band-split (analysis) stage, the 1/16th size picture (180 lines by 335 pels) is fed into a P × 64kbs coder. At the receiver, all of the sub-bands are combined by up-sampling and interpolation. The final HDTV pictures are reconstructed at the end of the synthesis stage.

We adopt the short length FIR filters suggested by LeGall *et al.* [6] for band-splitting and band-merging, since these filters are reported to have good performance for still picture compression [6]–[7]. In our simulations, the analysis (band-split) filter impulse responses are $[-1\ 2\ 6\ 2\ -1]$ and $[1\ -2\ 1]$ for the low-pass and the high-pass components, respectively. The corresponding synthesis filters (that allow perfect reconstruction if no coding errors and arithmetic errors are introduced) are given by the pair $[1\ 2\ 1]$ and $[1\ -2\ 6\ -2\ 1]$. The small number of taps of these filters is attractive for HDTV hardware implementation, even though longer filters with better frequency characteristics [8] could have been used.

We explored two channel allocation schemes. The first one uses a constant bandwidth assignment, and the second one

Band 1 1/4	Band 2 1/10	Band 5 1/10	Band 6 1/25
Band 3 1/10	Band 4 1/20	Band 7 1/25	Band 8 1/50
Band 9 1/10	Band 10 1/25	Band 13 1/25	Band 14 1/50
Band 11 1/25	Band 12 1/50	Band 15 1/50	Band 16 1/50

Fig. 10. Fixed channel allocation for each subband-coder.

uses the dynamic channel allocation algorithm described in Sec. III-B.

For most pictures, the lowest band contains most of the energy and should thus be allowed more bits for coding. Heuristicly, we set 1/4 of the total bandwidth to the lowest band. The two next higher bands, in the horizontal direction and in the vertical direction, are assigned 1/10th of the total channel bandwidth. A complete list of channel sharing factors is given in Fig. 10.

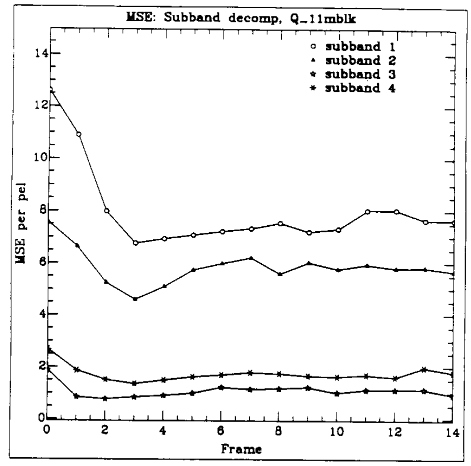
Fig. 11 shows the results of the lowest four bands, and Fig. 12 is the highest four bands. For this particular sequence, 1/4 of the total channel bandwidth is not sufficient for the lowest band, for it still has relatively high mean square errors and quantization step sizes, as compared to the higher bands. The distortions introduced by coding are visible at the lowest band, though they are not disturbing.

The reconstructed pictures after sub-band synthesis show more visible artifacts when they are compared to those generated by the spatial decomposition in Section III. The mean square errors in Fig. 13 also indicate the same observation.

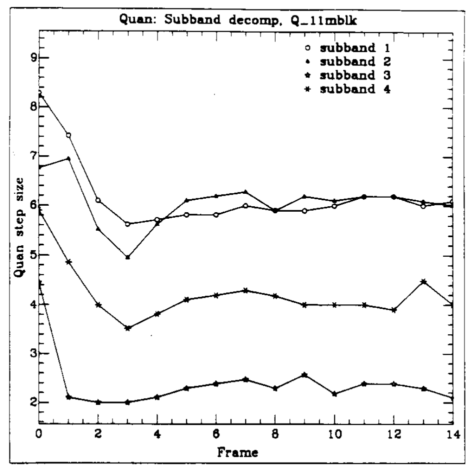
We also apply the dynamic channel allocation scheme to the sub-bands. The initial channel sharing factors are taken from the constant channel scheme as defined by Fig. 10. The simulation results are promising as shown in Figs. 14 and 15. However, because the average bit rates are extremely low at the highest few bands, the formula we use (8) is thus inaccurate in this situation (as discussed in Section III-B). Therefore, a ripple-like pattern can be seen on Fig. 15(b) and (c). The lower several bands, on the other hand, behave as we expect: the step sizes are compatible with each other, as well as their mean square errors.

The biggest advantage of this dynamic allocation scheme is that the lowest band has a much smaller distortion and that seems to be critical for the reconstructed pictures after sub-band synthesis (Fig. 13).

After the sub-band decomposition, each sub-band is quantized from a maximum of 23 bits dynamic range to eight bits, so as to be compatible with the $P \times 64$kbs specifications. This introduces some quantization errors to the sub-band signals prior to coding. Fig. 13 shows the mean square error introduced by this readjustment process (the "pre-Q" curve). Even without coding, minor visual artifacts can be detected on the synthesized pictures due to this readjustment. In general, the noise introduced at the sub-bands may become larger after the synthesis stage. A known solution to this problem is using the paraunitary filters [9] that preserve the $L2$ norm. That is, these filters guarantee that for stationary



(a)



(b)

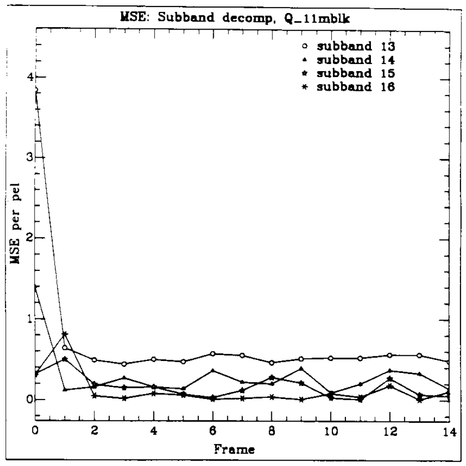
Fig. 11 (a) Lower bands mean square effort of sub-band decomposition using RM8—quantizer adjusted at every 11th macro-block. (b) Lower bands quantization step sizes of sub-band decomposition using RM8.

Gaussian sources and white noise, the noise power after reconstruction is the same as the noise power introduced at the sub-bands. Unfortunately, the linear-phase filter banks used in our experiments are not paraunitary.

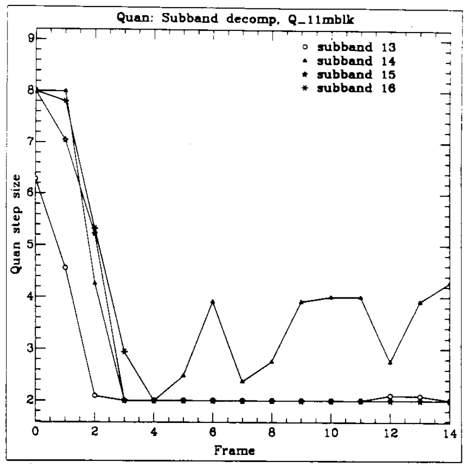
Two modifications thus can be incorporated to improve the performance of the sub-band approach:

1) Modify the sub-coders so that they can accept input data with a larger dynamic range (more than eight bits);
2) Choose a set of analysis/synthesis filters that do not amplify the quantization noise at the synthesis stage.

An even better performance could be expected by incorporating perceptually adjusted thresholds in encoding the various sub-bands as suggested in [10], [11]. These thresholds should be set, in each sub-band, according to the characteristics of the motion-compensated frame difference signals.



(a)



(b)

Fig. 12 (a) Higher bands mean square errors of sub-band decomposition using RM8—quantizer adjusted at every 11th macro-block. (b) Higher bands quantization step sizes of sub-band decomposition using RM8.

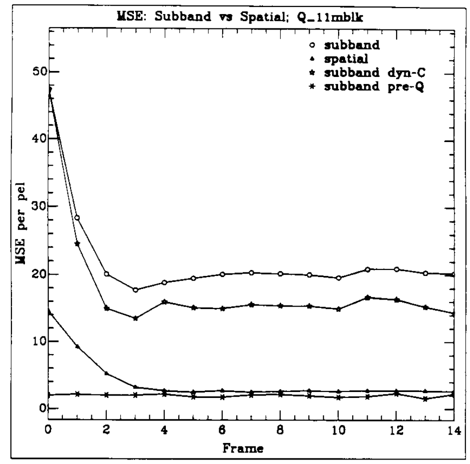


Fig. 13. Mean square errors of various reconstructed HDTV pictures.

V. Discussion

The high speeds necessary for processing HDTV signals make compression by traditional methods impractical. We have explored two methods, one in spatial domain and the other in frequency domain, that use parallel processing to distribute the video compression task among a number of coders. Our focus has been on the CCITT P × 64 kbs standard ISDN videotelephone codec, with the assumption that competition and high volume will make it economically attractive. Coders of this type are implementable with today's technology, and some are already available in the market.

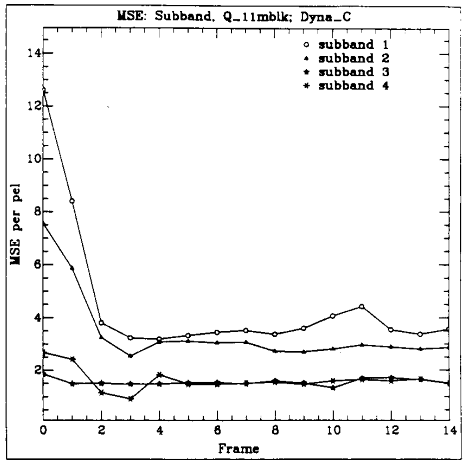
This argument could also be extended to the coder described in the emerging MPEG[12] standard, which has features that could be used to further reduce the bit-rate, and insure that a high quality picture is maintained. Of primary interest would be the reduction of temporal redundancy with motion compensated interpolation [13]. This technique introduces delay in the signal path, which causes difficulty in interactive communication, but for most HDTV applications the transmission is unidirectional, and a matching audio delay is all that would be needed to make the technique transparent. The other major disadvantage of the technique, which is the cost and complexity of the hardware, should be alleviated by virtue of the standardization.

HDTV image can be partitioned in either the spatial domain (spatial decomposition) or the frequency domain (sub-band decomposition). If we assume that no information is exchanged among the parallel processing sub-coders, a key problem with this coder structure would be the nonuniform picture quality caused by inappropriate bit rate assignment to the sub-coders, and the variation of the quantizer step size inside a sub-picture. Therefore, a dynamic channel bit allocation scheme and a quantizer adjustment scheme have been proposed to solve this problem. They are all based on a simple source signal model which approximates the relationship between the bits and the quantization step size of a coding process. Our simulations show very promising results using these proposed schemes.

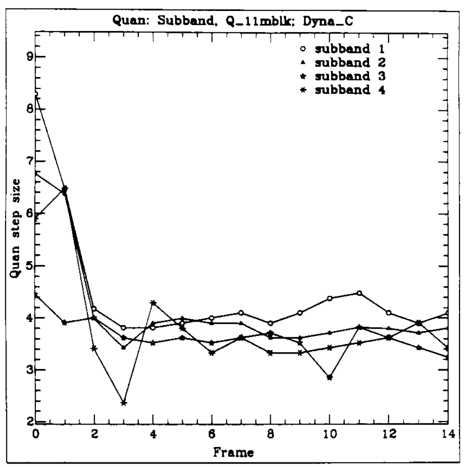
Our results for the spatially subdivided image show no degradation in picture quality under normal viewing conditions. The simulations on the sub-band coded images indicate that the RM8 coder may need modifications to produce better quality pictures. This may be due to the fact that the pre-quantization and coding noise introduced in the sub-bands are magnified at the reconstruction stage. Also, the motion compensation is done on the sub-band pictures with integer pel accuracy; this may significantly reduce its effectiveness.

Acknowledgment

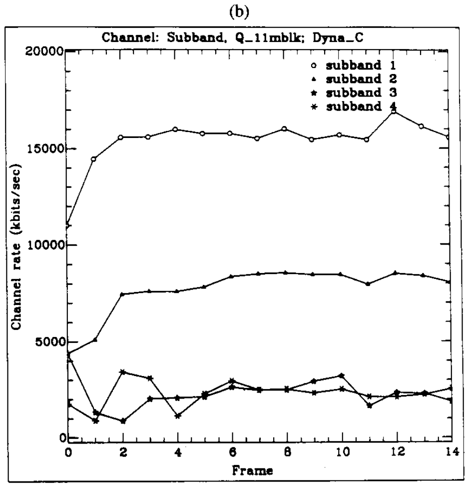
The authors would like to thank Richard Kollarits and David Gibbon for generating the test picture sequences used.



(a)

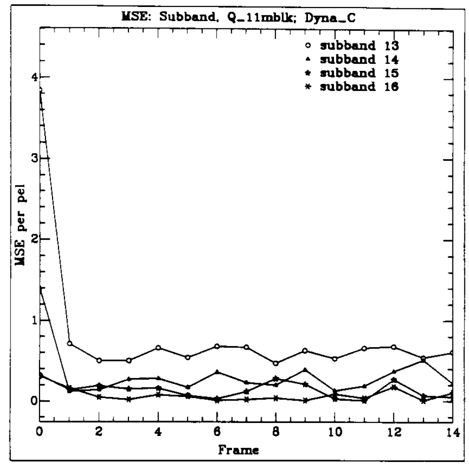


(b)

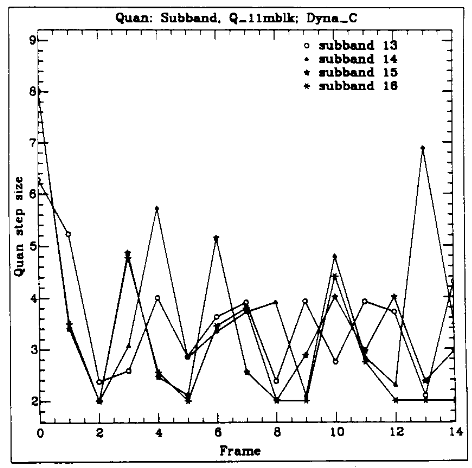


(c)

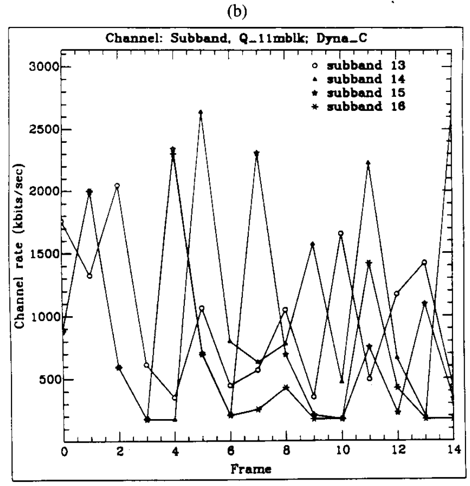
Fig. 14 (a) Mean square errors of dynamic channel allocation algorithm applied to sub-band coder, in lower bands. (b) Quantization step-sizes of dynamic channel allocation algorithm applied to sub-band coder, in the lower bands. (c) Channel bit rates of dynamic channel allocation algorithm applied to sub-band coder, in lower bands.



(a)



(b)



(c)

Fig. 15 (a) Mean square errors of dynamic channel allocation algorithm applied to sub-band coder, in higher bands. (b) Quantization step sizes of dynamic channel allocation algorithm applied to sub-band coder, in higher bands. (c) Channel bit rates of dynamic allocation algorithm applied to sub-band coder, in higher bands.

References

[1] T. S. Rzeszewski, "A technical assessment of advanced television," *Proc. IEEE*, pp. 789-804, May 1990.

[2] CCITT, Working Party XV/1, Draft of Recommendation H.261, "Video Codec for Audiovisual Services at px64 kbits/s," July 1990, see also, Recommendations H.221, H.242, H.230, and H.320.

[3] CCITT, Working Party XV/4, "Description of Reference Model 8 (RM8)", Specialists Group on Coding for Visual Telephony, Doc. No. 525, June 1989.

[4] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ: Prentice-Hall, 1984, p. 154.

[5] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation and Compression*. New York: Plenum, 1988, pp. 424-425.

[6] D. J. LeGall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *Proc. ICASSP'88 Conf.*, New York, Apr. 1988, pp. 761-764.

[7] T. Kronander, "Some aspects of perception based image coding," Linköping studies in science and technology," Ph.D. Dissertation no. 203, Dept. of Elec. Eng., Linköping Univ., Sweden, 1989.

[8] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. ICASSP'80 Conf.*, Apr. 1980, pp. 291-295.

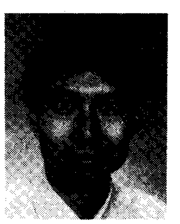
[9] M. Vetterli and D. J. LeGall, "Perfect reconstruction FIR filter banks: Some properties and factorizations," *IEEE Trans. Acoust., Speech, Signal Processing* vol. 37, no. 7, July 1989.

[10] R. J. Safranek, K. MacKay, N. S. Jayant, and T. Kim, "Image Coding based on selective quantization of the reconstruction noise in the dominant sub-band," in *Proc. ICASSP'88 Conf.*, New York, Apr. 1988, pp. 765-767.

[11] R. J. Safranek and J. D. Johnston, "A perceptually tuned sub-band image coder with image dependent quantization and post-quantization data compression," in *Proc. ICASSP'89 Conf.*, May 1989, pp. 1945-1948.

[12] Motion Picture Experts Group—ISO-IEC/JTC1/SC2/WG8/MPEG.

[13] A. Puri, R. Aravind, B. G. Haskell, and R. Leonardi, "Video coding with motion-compensated interpolation for CD-ROM applications," *Signal Processing: Image Communication*, July 1990.



Hsueh-Ming Hang (S'79-M'80-S'80-M'83-S'83-M'84-SM'90) was born in Taiwan in 1956. He received his B.S. degree in control engineering and M.S. degree in electronics from National Chiao-Tung University, Hsinchu, Taiwan, in 1978 and 1980, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1984.

From 1980 to 1983 he was a Graduate Assistant in the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute. From 1983-1984 he was an instructor in the same department; teaching courses in linear and communication systems. Since 1984 he has been with the Visual Communications Research Department, AT&T Bell Laboratories, Holmdel, NJ, where he is engaged in data compression algorithm and harddware researches for still and motion imagery. His current research interests include digital video compression, VLSI architectures for image processing, multidimensional signal processing, and information theory.

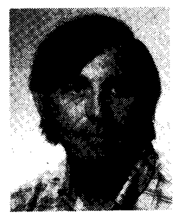
He has published over 20 technical papers on image compression, signal processing, and video codec architectures. He is a member of IEEE Circuits and System Society's Committee on Visual Signal Processing and Communications.

Riccardo Leonardi (S'80-M'83-S'84-M'84-S'85-M'87) received the Diploma and Ph.D. in electrical engineering from the Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland, in 1984, and 1987, respectively. In 1981, he spent one year at Carnegie-Mellon University, and in 1988, he was a Post-Doctoral Fellow at University of California, Santa Barbara.

Since 1988, he has been performing research at AT&T Bell Laboratories in the field of Visual Communications. In 1991, he was appointed to coordinate the research and teaching activities of the Signal Processing Laboratory of EPFL. He has taught courses in the fields of Digital Signal Processing, Image Processing and Video Communications and is a Contract Professor at Geneva University. His research interests include digital techniques for coding and transmission of video signals and images. He is also involved in the coordination and supervision of several European Economic Community projects for video compression.

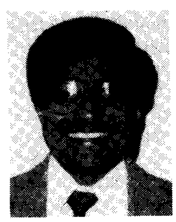
Dr. Leonardi is author and co-author of several international scientific publications, and a member of EURASIP.

Barry G. Haskell (S'65-M'68-SM'76-F'87), for a photograph and biography, see this issue, p. 170.



Robert L. Schmidt received the B.S.E.E. degree in 1982 from Monmouth College. Mr. Schmidt is currently a member of the Information Systems Research Laboratory at AT&T Bell Laboratories, Holmdel, NJ, where he is studying visual communications with large scale integration, and its applications. Previous work has included video signal compression for applications in teleconferencing and various methods for combining television signals for satellite communications.

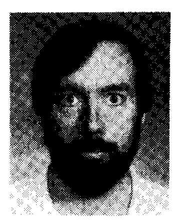
Mr. Schmidt is a member of the Eta Kappa Nu Society.



Hemant Bheda received the B.E. degree from Bombay University in 1982 and the M.S. degree from San Diego State University in 1984.

From 1984 to 1987, he worked as a VLSI Design Engineer for Intel Corporation, and was involved the development of the 80286 and 80386SX microprocessors. In 1987 he joined the VLSI systems research department at AT&T Bell Laboratories. He has been working in the area of VLSI architectures and algorithms for video compression.

Mr. Bheda is a recipient of the National Scholar Award from the Government of India, and a member of Eta Kappa Nu Society.



Joseph H. Othmer received the B.S. and M.S. degrees from Monmouth College in 1982 and 1987, respectively, both in electronic engineering.

He has been with AT&T Bell Labs., Holmdel, NJ since 1982. His research interests include video coding, VLSI, and CAD.