Abstract - A computationally simple algorithm for video compression is presented in this paper. The algorithm gives excellent quality image sequence reconstruction at a peak-signal-to-noise-ratio (PSNR) of approximately 37 decibels (dB) and a very low bit rate of 0.2 bits-per-pixel (bpp). The scheme utilizes absolute moment block truncation coding (AMBTC) for intraframe coding, in conjunction with three-step algorithm for block based motion compensation, and a simple quantization scheme for interframe error coding. The procedural steps in the algorithm are very similar to the MPEG standard, which can facilitate the implementation of the proposed algorithm in the existing infrastructure. The results of the simulations performed on three different image sequences depicting different amounts of motion and background activity are presented. The qualitative and quantitative comparisons of the proposed algorithm, with an existing video compression standard and with other similar research are presented. The significant advantage gained in terms of computational complexity and the probable use of the algorithm for progressive transmission of video is illustrated. The modifications of the basic algorithm are also presented in this paper.

I. INTRODUCTION

Several application specific standards for video compression are available. Out of them the one referred to in this paper is the most prevalent and widely used MPEG (Moving picture experts group) video standard [1, 2, 3 and 4]. MPEG is also an application specific standard and different versions of MPEG are available for different applications and bit rates. The basic algorithm for all these versions is the same and is very similar to the other video compression standards. In MPEG an image sequence is divided into groups of picture frames and each group is coded independently. For providing random access capabilities one frame in every group is coded independently and thus can be reconstructed at the receiver independently. Such a frame is called an I-frame (or independent frame). The rest of the frames are coded as P-frames or B-frames. The P-frame (or predicted frame) is coded using the last I-frame or P-frame whichever happens to be the closest. The I-frames and P-frames are called anchor frames. The B-frames (or bidirectionally-predicted frames) are predictive coded using the most recent anchor frame and the closest future anchor frame. The I-frames, and the error frames obtained after prediction, are coded using JPEG (Joint photographic experts group) algorithm [1, 2, 3 and 4]. JPEG is a well-known computationally intensive algorithm and thus MPEG also becomes a computationally intensive scheme.

In this paper the aim is to design a compression algorithm that is computationally simpler, faster and better in performance than the present video compression algorithms. The peak-signal-to-noise-ratio (PSNR) [1] and the bits-per-pixel (bpp) [1] requirements are taken as the main quantitative evaluation parameters and the visual quality of the reconstructed sequence is considered as the qualitative evaluation parameter.

The paper is organized as follows. In the next section the designed algorithm is introduced followed by a discussion on the simulation tests performed and the results obtained. The performance comparison of the proposed algorithm is made with similar works. The conclusions of the research work are also presented.

II. THE PROPOSED ALGORITHM

The central goal of this research work is to devise a simple, fast and high performance algorithm for video compression. The idea is to analyze the weaker points of the existing algorithms and substitute the complex building blocks of those algorithms by some simple schemes preserving the quality performance of the system but reducing the computational complexity significantly. As MPEG is the most widely used and one of the best performing algorithms, the structure of the MPEG algorithm is taken as a reference. The basic structure of the algorithm is preserved to ease the implementation of the new algorithm as a replacement for the existing counterparts with minimal change in supporting infrastructure. The algorithm introduced in this paper presents a first step in the inclusion of AMBTC [5] to video compression. The basic steps in the algorithm can be described as follows:

A sequence of N digitized image frames is divided into groups of k picture frames each, and each group is coded independently. To provide random access capability at the receiver, the first frame in every group is coded independently and can be reconstructed independently at the receiver. The coding scheme used for this intraframe coding is AMBTC [5]. The image frame is divided into n x n pixel blocks. For every block, the mean (\( \eta \)) and the first absolute moment (\( \alpha \)) are calculated as follows:
\[ \eta = \frac{1}{m} \left( \sum_{i=1}^{m} X_i \right) \]  

(1)

and

\[ \alpha = \frac{1}{m} \left( \sum_{i=1}^{m} (X_i - \eta) \right) \]  

(2)

where \( X_i \) is the pixel value and \( m = n \times n \) is the total number of pixels in a block. A bit-plane (\( \bar{X} \)) is obtained which contains ones in the places where \( X_i \geq \eta \) and zeros elsewhere. The mean, absolute moment and the bitmap are coded and transmitted. At the receiver for every block the low-mean \( (a) \) and the high-mean \( (b) \) are calculated as follows:

\[ a = \eta - \frac{\gamma}{m-q} \]  

(3)

and

\[ b = \eta + \frac{\gamma}{q} \]  

(4)

where \( q \) is the number of pixel values in a block greater than the mean and

\[ \gamma = \sum_{X_i \geq \eta} X_i - (\eta q) \]  

(5)

The pixels corresponding to a 1 in the bitmap are assigned the high-mean corresponding to that block, while the pixels corresponding to a 0 in the bitmap are assigned the low-mean to reconstruct the block at the receiver. AMBTC gives very good reconstruction and the coding can be performed at a bit rate of around 1.6 bpp using the techniques applied in [5]. This corresponds to the JPEG coded I-frame in MPEG.

The pixels corresponding to a 1 in the bitmap are assigned the high-mean corresponding to that block, while the pixels corresponding to a 0 in the bitmap are assigned the low-mean to reconstruct the block at the receiver. AMBTC gives very good reconstruction and the coding can be performed at a bit rate of around 1.6 bpp using the techniques applied in [5]. This corresponds to the JPEG coded I-frame in MPEG.

The first \( k \) images of the test sequence are grouped together and the algorithm is applied on them. The value of \( k \) is calculated. If the MAE is greater than a pre-decided threshold value, the error block is quantized and coded. If the MAE is smaller than the threshold value, the block is not coded and is not transmitted at all. The threshold can be obtained adaptively depending upon the bits-per-pixel requirements or the SNR requirements or both. The information about transmitting an error block, or not transmitting it, can be sent as a header bit with minimal increase in the bpp requirements. For coding the error block to be sent, a uniform quantizer [1] is used with the code assignment performed using pre-assigned Huffman code.

The previously reconstructed frame, motion vectors and the error blocks transmitted help in reconstructing the image frame at the receiver. This procedure is repeated for all the frames in a group and thereafter the whole scheme is iterated for all the groups in the sequence to reconstruct the whole image sequence at the receiver. The block diagram shown in Fig. 1 illustrates the steps of the proposed algorithm.

III. SIMULATION AND RESULTS

The simulation and testing of the proposed algorithm was performed on three popular video sequences. The sequences, their properties and their sizes are as follows:

- Miss America sequence – slow motion, scanty information – 288 x 360
- Susie sequence – faster motion, larger tonal change – 480 x 720
- Salesman sequence – fast motion, noisy background – 288 x 360

All the sequences were 256 level grayscale images with each pixel requiring 8 bits for representation. Synchronization and header information is excluded in the given size. 30 frames per second with non-interlaced type of frames are assumed for video transmission.

The first \( k \) images of the test sequence are grouped together and the algorithm is applied on them. The value of \( k \)
is varied from 1 to 15. The first frame in all the cases is AMBTC coded with a block size of (4 x 4) pixels. The frame is coded at 1.6 bpp using the adaptive bit assignment scheme discussed in [5]. Differentially encoding the mean and absolute moment values was also tried. Both the techniques give approximately the same bpp requirements. The PSNR achieved for all the sequences is over 37 dB and the image quality is very good.

For the rest of the frames, motion compensation is applied using the three-step algorithm. Using a block size of (16 x 16) pixels and pre-assigned Huffman coding, the motion vectors can be coded at 0.015 bpp.

For the error frame coding, the blocks, which contribute to the MAE greater than a threshold value, are identified. The threshold is varied between 4 – 8 adaptively depending upon the bpp requirements. A 16-level mid-rise uniform quantizer with a step size of 8 is used for coding the error values, and the code assignments are again made using pre-assigned Huffman code.

The reason for the choice of the upper limit of threshold values can be derived from the PSNR calculations. A PSNR of 35 dB is supposed to be a sufficient value of PSNR to support good quality image sequence reconstruction. If backtracking is performed using the PSNR formula, this corresponds to an average error value of over 4. Thus if a threshold of 8 is chosen for the MAE, assuming that the error will be distributed uniformly, it can be safely assumed that the average error will be around 4 and so a PSNR of over 35 dB can be expected, assuring a decent image reconstruction quality.

The quantitative results of the simulations performed for the Miss America sequence can be seen in the form of the PSNR versus number of frames curve (Fig. 2(a)) and the bpp versus number of frames curve (Fig. 2(b)) at different values of the MAE threshold. The performance curves for the other two sequences follow a similar pattern with a slight reduction in the PSNR and a slight increase in bpp requirements. It can be observed that decreasing the threshold improves the PSNR but the bpp is also increased. The typical values of the threshold, PSNR, MSE, bpp and bit-rate for the three sequences are shown in the Table I. All the three sequences give a very good PSNR, and a very low bit-rate. As an illustration of the excellent reconstruction quality, the first and eighth frames of the original and the reconstructed images of Miss America sequence for the typical parameter values are shown in Fig. 3. The figure clearly indicates that the reconstructed sequence resembles the original sequence very closely, proving the effectiveness of the proposed algorithm. The comparison of the results with other compression schemes is based on the Miss America sequence and so frames of only the Miss America sequence are shown.

IV. PERFORMANCE EVALUATION

The performance of the proposed algorithm is evaluated in two stages. Firstly, its performance is compared with the performance of MPEG standard algorithm and then it is compared with the other contemporary work done in this field.

A. Comparison with MPEG

The MPEG standard does not give detailed rules of how a sequence should be coded. The standard is like a protocol for the decoding procedure. The broad guidelines for achieving good compression are mentioned but many choices are left for the user to decide.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>MISS AMERICA (288*360)</th>
<th>SUSIE (480*720)</th>
<th>SALESMAN (288*360)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>13.9</td>
<td>16.71</td>
<td>19.64</td>
</tr>
<tr>
<td>PSNR</td>
<td>36.7 dBs</td>
<td>35.9 dBs</td>
<td>35.2 dBs</td>
</tr>
<tr>
<td>Bpp required</td>
<td>0.19</td>
<td>0.21</td>
<td>0.23</td>
</tr>
<tr>
<td>Bit rate</td>
<td>0.59 Mbps</td>
<td>2.18 Mbps</td>
<td>0.72 Mbps</td>
</tr>
</tbody>
</table>

Table I: Performance Characteristics for Different Test Sequences

Fig. 2. Performance curves for Miss America sequence.
(a) PSNR vs Number of frames and (b) bpp vs Number of frames.
Thus the comparison made here with MPEG is based on the simulation results obtained in past research work [8] and analytical comparison based on the basic algorithm in MPEG.  

1) Qualitative evaluation: The visual quality of the reconstructed sequence is excellent in both the algorithms. The reconstructed sequence is almost indiscernible from the original sequence.

2) Quantitative evaluation: The performance of the two algorithms is compared based on the signal to noise ratio, bits per pixel requirements, computational complexity and storage requirements.

i) PSNR: The average PSNR of the reconstructed Miss America sequence using MPEG is approximately 37 dB [8]. As can be seen from Fig. 2(a), a similar PSNR can be achieved with the algorithm presented in this paper with a threshold of 7 for the MAE. The proposed algorithm presents a major advantage over MPEG, by supporting the possibility of progressive transmission of the video signal. The threshold value of MAE for the coding of error blocks can be reduced recursively for continual refinement of any image frame at the receiver, providing progressive transmission capability. The information required to achieve such capability can be attached as header bits in front of the error coded blocks with minimal increase in the bits per pixel requirements.

ii) Bits per pixel (bpp) requirements: The bpp requirements for both the algorithms are very similar. The independently coded frame in MPEG is coded at around 1.2 bpp. The algorithm introduced in this paper codes the independently coded frame using AMBTC at 1.6 bpp. The coding of the motion vectors can be done more efficiently for the algorithm introduced in this paper, because the number of different motion vectors possible is very less when compared to the MPEG algorithm. For error coding in the proposed algorithm, only 2% of the total error blocks on an average are coded, at less than 2.5 bpp. Thus if we have to code the error blocks in MPEG to achieve similar average bpp requirements we have an average of approximately 3 bits per (8 x 8) block of pixels. In MPEG the DC coefficients and the AC coefficients are coded separately based on pre-assigned Huffman codes. On observing the sample tables for the pre-assigned Huffman codes [1 and 2], it can be safely claimed that the algorithm introduced in this paper is as efficient as the MPEG algorithm in terms of the bpp requirements.

iii) Computational complexity: Addition, multiplication, comparison and coding are considered as individual computational entities. The computations required for coding and comparisons can be neglected as the order of these operations is very nearly the same in both the algorithms. Both the algorithms under discussion can be divided into three different stages of operation: independent frame coding, motion compensation, and error coding. In MPEG for independent frame coding an approach similar to JPEG is used, and it gives computational requirements of approximately \(1.3 \times 10^7\) additions and a similar number of multiplications for an image frame of size (288 x 360). For the independent frame coding using AMBTC in the proposed algorithm, for an image frame of size (288 x 360), the computational requirements are approximately \(3.1 \times 10^5\) additions and \(2.6 \times 10^4\) multiplications. Motion compensation in case of MPEG requires 1022 x \((2n + 1)^2\) addition operations, where \(n\) is the length of the search region in any dimension, for every (16 x 16) block of data in a predictive coded frame. For a bi-directionally predictive coded frame, twice of this amount is required. In the proposed algorithm, for every (16 x 16) block of data, 1022 x 27 additions are required. Considering the fact that typically the number of bi-directionally coded frames in a sequence is more than twice the number of predictive coded frames, and typical values of \(n\) are around 10, the enormous amount of saving in computations in the proposed algorithm can be understood. To find an error frame 1.8 \(\times\) 10^3 additions are required for both the algorithms. For error coding in MPEG the procedure and hence the number of computations is the same as for the independently coded frame. For the proposed algorithm, after neglecting the comparison and encoding operations and taking into account that only 2% of the error blocks are coded, approximately 2 \(\times\) 10^5 additions and 2 \(\times\) 10^3 multiplications are required. For hardware implementation 48 gates are required for an 8-bit full adder and 496 gates are required for an 8-bit multiplier. If the total number of addition and multiplication operations required for both the algorithms are compared in view of this fact, the huge advantage of the proposed algorithm over MPEG can be truly appreciated.

iv) Storage requirements: In MPEG two frames are stored in the memory at any time while in the proposed algorithm only one frame is required. This increases the rate at which the memory has to be refreshed but the...
refreshment operation can be performed in parallel to the other computations on a block-by-block basis, thus preserving the advantage obtained by reduction in storage requirements.

B. Comparison with other work

The performance of the proposed algorithm is compared with the results obtained in related works in [9], [10] and [11] and the comparison is presented in Table II. Healy and Mitchell [9] use block truncation coding (BTC) for both interframe and intraframe coding. BTC is also a moment preserving code. The algorithm is very simple, but the reconstructed image quality is not very good. The PSNR is less (32 dB) and the bpp requirement is very high (0.9). Karlsson and Vitterli [10], and Nehal [11] have suggested schemes for progressive transmission using temporal filtering using symmetric short kernel filters (SSKFs). The reconstruction quality is good but the computational complexity is very high. The algorithm suggested in this paper gives better PSNR and required bpp, than the other three algorithms with lesser computational complexity.

V. CONCLUSIONS AND DISCUSSION

In this paper, a compression algorithm is presented that is computationally simpler, faster, and equally efficient if not better than the existing video compression standards in terms of quantitative as well as qualitative performance. The results obtained from the simulations are very encouraging and the use of AMBTC with some basic techniques in video compression like motion compensation and error coding has indeed helped in reducing the complexity involved in compressing the image sequence data.

AMBTC especially lends itself very effectively for video coding, as the artifacts generated by AMBTC algorithm don’t show up in the video sequences. The reason is that AMBTC conserves the lower order moments, which is similar to conserving the lower order frequencies, and the higher frequency artifacts are masked in image sequences anyway.

Thus while the strengths of AMBTC lead to significant reduction in the computational requirements, we don’t loose

in terms of the video quality for a human observer.

Coding the image sequences in their sequence of arrival helps in making the system closer to real time. The bit stream order and the processing order are same in this case and we don’t have to wait for the arrival of future anchor frames to perform motion compensated prediction and reconstruction. As only the previous reconstructed frame is used for prediction, the amount of storage required is reduced to half in comparison to the MPEG and the computational requirements are also reduced significantly.

In MPEG, the reconstructed future anchor frame is used in addition to the previous reconstructed anchor frame for coding the current frame to take care of any sudden appearance or disappearance at the scene known as occlusion effect and hence reduce the bit requirements to perform the coding. The occlusion effect doesn’t cause any special problems in the scheme described here because any sudden appearance or disappearance has to be coded once in any case. In MPEG this is done for the future frame while here we do it for the current frame. Thus without loosing much on the bit requirement, the storage requirements and computations required can be reduced as compared to MPEG. Thus the algorithm introduced here presents a very simple but effective solution to the problem of video compression.

The proposed algorithm can be easily modified for progressive transmission of video signals.

REFERENCES


<table>
<thead>
<tr>
<th>Table II</th>
<th>PERFORMANCE COMPARISON FOR DIFFERENT COMPRESSION METHODS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interframe Coding</td>
<td>Block truncation Coding (BTC)</td>
</tr>
<tr>
<td>Intraframe Coding</td>
<td>BTC</td>
</tr>
<tr>
<td>Average PSNR</td>
<td>32 dB</td>
</tr>
<tr>
<td>Bpp</td>
<td>0.9</td>
</tr>
<tr>
<td>Relative Complexity</td>
<td>Low</td>
</tr>
<tr>
<td>Comments</td>
<td>Moderate quality at a high bit rate</td>
</tr>
</tbody>
</table>