CHAPTER - 3

MODIFIED CODEC

As from the results of previous chapter, we can see that the compression ratio achieved using normal MPEG-2 is in the range of 9-12%. If we want higher compression ratio MPEG-4, giving compression ratio around 5%, is to be used, but the problem with MPEG-4 is higher computational complexity. MPEG-4 is object based video compression technique, which first of all separates moving objects from the non-moving objects. Afterwards, the moving objects are put in foreground and non-moving objects are put in background of the frame. Then these both frames are compressed separately from each other and the bit stream is generated for the transmitter. So due to parallel processing of two frames the codec for MPEG-4 is having higher computational complexity.

So for getting better compression ratio maintaining lower complexity of the codec, we need to make some modification in MPEG-2 codec. Here I have done modification in transform coding part of the codec, replacing only DCT by hybrid transformation which is combination of three transforms. Here in the first step truncated singular value decomposition (truncated SVD) is used to reduce the size of the residual frame. Then 2-D discrete wavelet transform (DWT) is taken of the truncated frame, which leads to four components LL, LH, HL and HH of the truncated frame. Then block wise discrete cosine transform (DCT) is taken of LL component.

The other all the steps in the codec are kept as they were, so the complexity of the codec is increasing slightly, but the improvement in compression ratio is achieved. The compression ratio achieved is nearer to MPEG-4 i.e. 5%, maintaining the quality of the video same, which can be seen from the results shown later in thesis.

3.1 Modified Video Codec:

The block diagram of the modified MPEG-2 based video codec (encoder and decoder) is as shown in the figure 3.1 and 3.2. The step wise working of the encoder and decoder is also explained.
3.1.1 Video Encoder:

Fig. 3.1: Block diagram of modified MPEG Video encoder

The working of the encoder is as below:

- First of all the video is converted into the sequence of frames which are nothing but like still images sequence.
- In next step the 10 sequential frames are selected (GOP) and format of the sequence is changed from RGB to YCbCr.
- Third step will be to resample the chrominance components of the frames from 4:4:4 to 4:2:0, as human eyes are less sensitive towards the chrominance components we can reduce no. of samples in that for compression.
- Then in group of pictures (GOP) the sequence is converted into sequence of IBBPBBPBBBI with use of motion estimation and compensation technique and residual frames are found out.
- Then all the residual frames are passed through truncated SVD to reduce size of frames.
- The next step is to take DWT of the motion compensated frames, which gives us four components namely: Approximate, Horizontal, Vertical and Diagonal.
- Out of these four components only approximate component is transmitted so DCT of it is taken.
The coefficients of the DCT of image are quantized using the standard quantization matrix.

The quantized coefficients of the transformed image are scanned in zig-zag manner.

The encoding process of these coefficients are done using run length coding for the better compression and the bit stream of the coder is stored on any medium or transmitted.

At last the bit-stream is formed and transmitted.

### 3.1.2 Video Decoder:

![Block diagram of modified MPEG Video decoder](image)

The working of the decoder is as explained below:

- In the first step the bit stream of MPEG is decoded using run length decoder and converted to original coefficients of DCT.
- The coefficients of DCT are quantized using quantization matrix to original DCT coefficients.
- Then inverse DCT is taken which will give approximate component of frame.
After appending zeros for horizontal, vertical and diagonal components of frame inverse DWT is performed.

Then the frames are motion compensated through motion estimation algorithm.

Then the chrominance component is re-sampled to original form.

The frames are converted back to RGB from YCbCr.

Finally the sequence of frames is converted to video which is final compressed video.

### 3.2 Truncated Singular Value Decomposition (SVD):

Singular value decomposition (SVD) is an effective tool for minimizing data storage and data transfer in the digital community. SVD transformation is a factorization of any frame into three parts. If the frame $A$ is applied with SVD, then it give us two orthonormal matrices $U$ and $V$, which contain eigenvectors of frame $A$ and third matrix as $S$ is a diagonal matrix, which contains Eigen values of corresponding eigenvector of frame $A$.

$$A = USV^T$$

Where,

$$U = [u_1 \ldots u_n], S = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}, V^T = \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$

If the size of $A$ is $m \times n$, then matrix $U$ and $V$ will be of the size $m \times m$ and $n \times n$ respectively, and $S$ will be of size $m \times n$ but the off diagonal values of $S$ will be zero always. Here the singular values of frame $A$ in $S$ matrix must fulfill the condition as:

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_r \geq \ldots \geq \lambda_n \geq 0$$

Where $r$ is a rank of frame $A$. Out of these entire higher ranked singular values are having so much lower values and lesser weightage in the reconstructed frame so we can remove that values for the compression of the frame. If we use only $k$ number of singular values, then the eigenvectors required to store and transmit the frame are also lesser.

$$\tilde{A} = \sum_{i=1}^{k} u_i s_i v_i^T$$

55
So truncated SVD transform removes higher order singular values and vectors, which only transmits \( k \) numbers of singular values and vectors out of all \( n \) number of values. So the recovered frame will require much lesser bits to store and transmit the same frame compared to original one. Fig. 3.1 shows the original and truncated image as a sample.

![Fig. 3.3: Effect of Truncated SVD (a) original frame (b) truncated frame](image)

As we can understand that the value of \( k \) controls the quality of the image, as we are going to reduce the value of \( k \) the bits required to represent the frame will reduce but the quality will decrease. As we are going to increase the value of \( k \), the quality will improve but the number of bits required will also increase. So the trade-off between quality and number of bits required is to be maintained using \( k \).

### 3.3 Discrete Wavelet Transform (DWT):

Discrete wavelet transform is a very important category of the sub-band coding. A 2-D DWT transforms a frame of video from the spatial domain to the frequency domain. The wavelet transform provides a time-frequency representation of the signal, which uses multi resolution technique to analyze different frequencies with different resolutions. Wavelets are localized waves which have energy concentrated in time or space, so that they are suitable to analyze transit signals.

In this process, time scale representation of the digital signal is obtained using digital filtering techniques. The signal is passed through low pass and high pass filters, out of which low pass filter provides approximation of the signal and high pass filters gives detailed information of the signal.
Output of DWT for a MxN frame contains four sub-bands, LL, LH, HL and HH, each having the size M/2 x N/2. Here LL component is approximation of the original frame, LH and HL represent horizontal and vertical coefficients of frame and HH component represent diagonal coefficients of the frame. LL band contains most of the energy of the frame and energy reduces as we go from LL to LH to HL to HH band. Any of this band can further be applied DWT to be reduced into further four sub-bands. DWT does have the excellent spatio-frequency localization property and thus it is very useful for video compression.

**3.4 Results:**

Here the modified algorithm is also implemented in MATLAB version 7.10.0.499a, as per the steps explained above in the video encoder and decoder sections. Testing of the algorithm is also done with the same five standard videos which are having
differences in the motion, color component etc. from each other and having size of 128 x 128 pixels to make study uniform.

Figure 3.6 shows compressed video frames of the digital video named, ‘foreman.avi’. Here thirty frames are shown as the video is having frame rate equal to 30 frames/second.

The compression ratio for the video ‘foreman.avi’ is as below:

\[
\text{Compression ratio} = \frac{\text{Compressed Video Data}}{\text{Original Video Data}}
\]

So from this equation,

\[
\text{Compression ratio} = \frac{(8106 + 8443 + 7559) * 2 + 73728 + 54}{128 * 128 * 3 * 30} = 0.0577
\]

The above calculation is including the following details: the final coefficients to be transmitted after run length encoding, coefficients of motion vector and flag bits for bi-directionally predicted frames in the numerator and in denominator number of pixels in 1 second color video with size of 128 x 128.
Now, the average values of subjective quality measurement parameters are as shown in below table:

**Table 3.1: Average values of parameters for ‘forman.avi’**

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>MSE</th>
<th>SSIM</th>
<th>MSAD</th>
<th>VQM</th>
<th>Blocking Beta</th>
<th>Blurring Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>22.25</td>
<td>387.26</td>
<td>0.6266</td>
<td>13.036</td>
<td>5.1864</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.15</td>
<td>11.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24.52</td>
<td>24.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3.6: Compressed Video frames (foreman.avi)
All the above parameters are averaged over 30 frames, and their frame wise values are as shown in the figure 3.7.

(a) Peak Signal to Noise Ratio

(b) Mean Square Error
(c) Structural Similarity

(d) Mean Sum of Absolute Difference
(e) Video Quality Matrices

(f) Blocking Matrices
(g) Blurring Matrices

Figure 3.8 shows compressed video frames of the digital video named, ‘vipmen.avi’. Here thirty frames are shown as the video is having frame rate equal to 30 frames/second.

The compression ratio for the video ‘vipmen.avi’ is as below:

\[
\text{Compression ratio} = \frac{\text{Compressed Video Data}}{\text{Original Video Data}}
\]

So from this equation,

\[
\text{Compression ratio} = \frac{(4342 + 4872 + 6415) * 2 + 73728 + 54}{128 * 128 * 3 * 30} = 0.0495
\]

The above calculation is including the following details: the final coefficients to be transmitted after run length encoding, coefficients of motion vector and flag bits for bi-directionally predicted frames in the numerator and in denominator number of pixels in 1 second color video with size of 128 x 128.
Fig. 3.8: Compressed Video frames (vipmen.avi)

Now, the average values of subjective quality measurement parameters are as shown in below table:

Table 3.2: Average values of parameters for ‘vipmen.avi’

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>MSE</th>
<th>SSIM</th>
<th>MSAD</th>
<th>VQM</th>
<th>Blocking Beta</th>
<th>Blurring Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Original</td>
<td>Original</td>
</tr>
<tr>
<td>21.667</td>
<td>463.88</td>
<td>0.7553</td>
<td>10.177</td>
<td>5.7058</td>
<td>15.007</td>
<td>15.757</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Compressed</td>
<td>Compressed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.797</td>
<td>19.1939</td>
</tr>
</tbody>
</table>

All the above parameters are averaged over 30 frames, and their frame wise values are as shown in the figure 3.9.
(a) Peak Signal to Noise Ratio

(b) Mean Square Error
(c) Structural Similarity

(d) Mean Sum of Absolute Difference
(e) Video Quality Matrices

(f) Blocking Matrices
Figure 3.10 shows compressed video frames of the digital video named, ‘viplane.avi’. Here thirty frames are shown as the video is having frame rate equal to 30 frames/second.

The compression ratio for the video ‘viplane.avi’ is as below:

\[
\text{Compression ratio} = \frac{\text{Compressed Video Data}}{\text{Original Video Data}}
\]

So from this equation,

\[
\text{Compression ratio} = \frac{(5674 + 6617 + 5833) + 73728 + 54}{128 \times 128 \times 3 \times 30} = 0.0526
\]

The above calculation is including the following details: the final coefficients to be transmitted after run length encoding, coefficients of motion vector and flag bits for bi-directionally predicted frames in the numerator and in denominator number of pixels in 1 second color video with size of 128 x 128.
Now, the average values of subjective quality measurement parameters are as shown in below table:

**Table 3.3: Average values of parameters for ‘viplane.avi’**

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>MSE</th>
<th>SSIM</th>
<th>MSAD</th>
<th>VQM</th>
<th>Blocking Beta</th>
<th>Blurring Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>Compressed</td>
<td>Original</td>
<td>Compressed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.189</td>
<td>622.42</td>
<td>0.5414</td>
<td>13.043</td>
<td>6.9409</td>
<td>12.673</td>
<td>11.733</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.81</td>
<td>17.986</td>
</tr>
</tbody>
</table>

All the above parameters are averaged over 30 frames, and their frame wise values are as shown in the figure 3.11.
(a) Peak Signal to Noise Ratio

(b) Mean Square Error
(c) Structural Similarity

(d) Mean Sum of Absolute Difference
(e) Video Quality Matrices

(f) Blocking Matrices
Figure 3.12 shows compressed video frames of the digital video named, ‘viplanedeparture.avi’. Here thirty frames are shown as the video is having frame rate equal to 30 frames/second.

The compression ratio for the video ‘viplanedeparture.avi’ is as below:

\[
Compression\ ratio = \frac{Compressed\ Video\ Data}{Original\ Video\ Data}
\]

So from this equation,

\[
Compression\ ratio = \frac{(6999 + 5897 + 5520) \times 2 + 73728 + 54}{128 \times 128 \times 3 \times 30} = 0.0485
\]

The above calculation is including the following details: the final coefficients to be transmitted after run length encoding, coefficients of motion vector and flag bits for bi-directionally predicted frames in the numerator and in denominator number of pixels in 1 second color video with size of 128 x 128.
Now, the average values of subjective quality measurement parameters are as shown in below table:

Table 3.4: Average values of parameters for ‘viplanedeparture.avi’

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>MSE</th>
<th>SSIM</th>
<th>MSAD</th>
<th>VQM</th>
<th>Blocking Beta</th>
<th>Blurring Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Original</td>
<td>Compressed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Original</td>
<td>Compressed</td>
</tr>
<tr>
<td>23.5</td>
<td>290.80</td>
<td>0.7338</td>
<td>8.166</td>
<td>4.1163</td>
<td>16.5</td>
<td>10.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9.445</td>
<td>12.3205</td>
</tr>
</tbody>
</table>

All the above parameters are averaged over 30 frames, and their frame wise values are as shown in the figure 3.13.
(a) Peak Signal to Noise Ratio

(b) Mean Square Error
(c) Structural Similarity

(d) Mean Sum of Absolute Difference
Figure 3.14 shows compressed video frames of the digital video named, ‘viptraffic.avi’. Here thirty frames are shown as the video is having frame rate equal to 30 frames/second.

The compression ratio for the video ‘viptraffic.avi’ is as below:

\[
\text{Compression ratio} = \frac{\text{Compressed Video Data}}{\text{Original Video Data}}
\]

So from this equation,

\[
\text{Compression ratio} = \frac{(4584 + 5386 + 8115) \times 2 + 73728 + 54}{128 \times 128 \times 3 \times 30} = 0.0543
\]

The above calculation is including the following details: the final coefficients to be transmitted after run length encoding, coefficients of motion vector and flag bits for bi-directionally predicted frames in the numerator and in denominator number of pixels in 1 second color video with size of 128 x 128.
Now, the average values of subjective quality measurement parameters are as shown in below table:

Table 3.5: Average values of parameters for ‘viptraffic.avi’

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>MSE</th>
<th>SSIM</th>
<th>MSAD</th>
<th>VQM</th>
<th>Blocking Beta</th>
<th>Blurring Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Original</td>
<td>Original</td>
</tr>
<tr>
<td>18.67</td>
<td>884.14</td>
<td>0.5525</td>
<td>17.424</td>
<td>7.214</td>
<td>13.09</td>
<td>21.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Compressed</td>
<td>Compressed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11.679</td>
<td>21.119</td>
</tr>
</tbody>
</table>

All the above parameters are averaged over 30 frames, and their frame wise values are as shown in the figure 3.15.
(a) Peak Signal to Noise Ratio

(b) Mean Square Error
(c) Structural Similarity

(d) Mean Sum of Absolute Difference
(e) Video Quality Matrices

(f) Blocking Matrices
In this chapter, the modification in MPEG video codec is discussed. The transform coding of normal MPEG video codec is containing only block wise DCT of the frame, but here modification suggests hybrid transform coding technique comprising of three stages. In first step take truncated SVD of the frame, which reduces some amount of data to represent the same frame and helps to increase number of zeros in the final coefficients before transmission. In second step take 2-D DWT of the frame which gives four components LL, LH, HL and HH. In third step take DCT of LL component only. Due to the modification suggested, the compression ratios are increased in the range of 4 to 6%, which is almost half than the normal MPEG video codec.

The other results, like SSIM, Blocking Beta and Blurring Beta shows that the quality of the video is little decreased due to higher compression, which is as expected. The quality of the video can be increased by using some post processes like filtering, which can reduce the blocking and blurring artifacts. The PSNR for almost all the videos is around 20 dB, which is above limit of visibility defined by human visual system.