CHAPTER 3

DEVELOPMENT OF EFFICIENT MOTION ESTIMATION ALGORITHM FOR LOSSLESS VIDEO COMPRESSION

3.1 INTRODUCTION

Lossless video coding is useful for many applications, such as telemedicine and high quality multimedia systems. Most of the video compression techniques which exploit temporal redundancy are lossy compression. Lossless image compression algorithms allow the exact original image to be reconstructed from the compressed one. Lossy methods accept some loss of data in order to achieve higher compression. But, lossless compression is used when it is important that the original and the decompressed data be identical. Medical video images are frequently transmitted using lossless compression. In this chapter, a technique for lossless video sequence compression using wavelet based adaptive motion estimation and predictive coding is presented as a method to improve the compression efficiency. In this work, the motion vectors are adaptively estimated in spatial and wavelet domain based on inter frame similarity using correlation approach.

3.2 WAVELET BASED COMPRESSION

Most natural images have smooth color variations, with the fine details being represented as sharp edges in between the smooth variations. Technically, the smooth variations in color can be termed as low frequency
variations and the sharp variations as high frequency variations. The low frequency components constitute the base of an image, and the high frequency components add upon them to refine the image, thereby giving a detailed image. Hence, the smooth variations are demanding more importance than the details. Separating the smooth variations and details of the image can be done in many ways. One such way is the decomposition of the image using a Discrete Wavelet Transform (Hilton 1994).

Discrete Wavelet Transform (DWT) is the most popular transform for image based application. Wavelet analysis is an extremely powerful data representation method that allows the separation of images into frequency bands without affecting the spatial locality (Hilton 1994, Xian et al 1999, Sonj et al 2001, Park et al 2004). A 2-dimensional wavelet transform is applied to the original image in order to decompose it into a series of filtered sub-band images. In a discrete wavelet transform, an image can be analyzed by passing it through an analysis filter bank followed by a decimation operation. This analysis filter bank, which consists of a low pass and a high pass filter at each decomposition stage, is commonly used in image compression. When a signal passes through these filters, it is split into two bands. The low pass filter, which corresponds to an averaging operation, extracts the coarse information of the signal. The high pass filter, which corresponds to a differencing operation, extracts the detailed information of the signal. The output of the filtering operations is then decimated by two.

The wavelet transformation has two clear benefits compared to the conventional DCT-based compression methods. Firstly, low and high pass filtered parts are done separately, which enables frequency analysis during compression. Secondly, the small sub-images can be used in image processing, which results from the nested transformations. This enables hierarchical image analysis whereas only partly reconstructing the image (Kutila and Viitanen 2004).
A two-dimensional transform can be accomplished by performing two separate one-dimensional transforms. First, the image is filtered along the x-dimension using low pass and high pass analysis filters and decimated by two. Low pass filtered coefficients are stored on the left part of the matrix and high pass filtered on the right. Because of decimation, the total size of the transformed image is same as the original image. Then, it is followed by filtering the sub-image along the y-dimension and decimated by two. The two dimensional wavelet transform is achieved by implementing a bank of one-dimensional low-pass and high-pass analysis filters. For one level of decomposition, the image is decomposed into four orthogonal sub bands: LL, HL, LH, and HH as shown in Figure 3.1.

![Wavelet Decomposition Diagram](image)

**Figure 3.1 Wavelet Decomposition**

At the top left of the image is a low-pass filtered version of the original and moving to the bottom right, each component contains progressively higher-frequency information that adds the detail of the image. Wavelet decomposition at one-level has the three detail images, Low-High (LH), High-Low (HL), and High-High (HH), correspond to distinct frequency bands. The HL sub-band contains horizontal oriented features. Deductively, the LH sub-band contains vertically oriented structures, and the HH sub-band
contains diagonal structures. The LL sub-band is the low pass filtered version of the image and is further decomposed in the same manner, in the next octave. This collection of sub-images forms a multiresolution representation that organizes the image into a set of details appearing at different resolutions. Finally, the image has been split into four bands denoted by LL, HL, LH, and HH, after one level of decomposition. The LL band is again subject to the same procedure. This process of filtering the image is called pyramidal decomposition of image. The reconstruction of the image can be carried out by reversing the above procedure and it is repeated until the image is fully reconstructed.

It is noted that the high frequency components are relatively sparse and many of the coefficients in these components are zero or insignificant. The wavelet transform is thus an efficient way of de-correlating or concentrating the important information into a few significant coefficients. The wavelet transform is particularly effective for still image compression and has been adopted as part of the JPEG 2000 standard (Iain and Richardson 2003, Kutila and Viitanen 2004) and for still image texture coding in the MPEG-4 standard. Wavelet based coding (Xiang et al 1999) provides substantial improvements in picture quality at higher compression ratios. Over the past few years, a variety of powerful and sophisticated wavelet based schemes for image compression, have been developed and implemented. Because of the many advantages, wavelet based compression algorithms are the suitable candidates for the new JPEG-2000 standard (Taubman and Marcellin 2001). Because of their superior energy compaction properties and correspondence with the human visual system, wavelet compression methods have produced superior objective and subjective results (Xiang et al 1999, Sonja et al 2001). Sudhakar et al (2005) have pointed out that the wavelet based image compression methods have resulted in practical advances such as: superior low-bit rate performance, continuous–tone and bit-level compression, lossless and lossy compression, progressive transmission by pixel, accuracy and resolution, region of interest coding and others.
3.3 PROPOSED ADAPTIVE MOTION ESTIMATION

An efficient adaptive motion estimation and prediction for lossless video sequence compression has been implemented in this work. It exploits amount of temporal redundancies and adaptively selects the best prediction mode based on motion information. The successive frames are first correlated and temporal prediction in wavelet domain is applied into the frames that have less motion features and frames that are highly correlated are predicted in spatial domain. The block diagram of proposed method is shown in Figure 3.2.

![Figure 3.2 Flow diagram of the proposed method](image-url)
3.3.1 Adaptive Domain Selection

This step aims to determine the operational mode of video sequence compression according to its motion characteristics. The candidate operational modes are spatial domain and wavelet domain. The wavelet domain is extensively used for compression due to its excellent energy compaction. However, Gong et al (2004) have pointed out that motion estimation in the wavelet domain might be inefficient due to shift invariant properties of wavelet transform. Hence, it is unwise to predict all kinds of video sequences in the spatial domain alone or in the wavelet domain alone. Hence a method is introduced to determine the prediction mode of a video sequence adaptively according to its temporal redundancies. The amount of temporal redundancy is estimated by the inter-frame correlation coefficients of the test video sequence. The inter-frame correlation coefficient between frames can be calculated by Equation (3.1) (Ying Li and Khalid Sayood 2007). If the inter frame correlation coefficients are smaller than a predefined threshold, then the sequence is likely to be a high motion video sequence. In this case, motion compensation and coding the temporal prediction residuals in wavelet domain would be inefficient; therefore, motion estimation operations are applied on the sequence in the spatial mode. That is, those sequences that have larger inter frame correlation coefficients are predicted in direct spatial domain. The frames that have more similarities with very few motion changes are coded using temporal prediction in integer wavelet domain.

\[
C_{i,i+1} = \frac{\sum x \sum y (p_i(x,y) - \overline{p_i}) \cdot (p_{i+1}(x,y) - \overline{p_{i+1}})}{\sqrt{\left(\sum x \sum y (p_i(x,y) - \overline{p_i})^2 \cdot \sum x \sum y (p_{i+1}(x,y) - \overline{p_{i+1}})^2\right)}}
\]  

(3.1)
where $C_{i,i+1}$ is the correlation coefficient between $i^{th}$ and $i+1^{th}$ frames, $p_i(x,y)$ and $\bar{p}_i$ are individual pixel values and mean pixel value in the current frame ($i^{th}$ frame) respectively and $p_{i+1}(x,y)$ and $\bar{p}_{i+1}$ are individual pixel values and mean pixel value in the next frame ($i+1^{th}$ frame) respectively.

Another preprocessing step in this phase is the conversion of RGB Color space into YUV color space. This is mainly done for reducing the color redundancy if the image is in RGB. The conversion is implemented by applying the following reversible color transform (Taubman and Marcellin 2001) as defined in Equation (3.2) and (3.3).

\[
\begin{align*}
Y &= 0.257 \; R + 0.504 \; G + 0.098 \; B + 16 \\
U &= -0.439 \; R - 0.368 \; G - 0.071 \; B + 128 \\
V &= -0.148 \; R - 0.291 \; G + 0.439 \; B + 128
\end{align*}
\] (3.2)

where

- $Y$ is the color as brightness
- $U$ is Blue – $Y$ (luma)
- $V$ is Red – $Y$ (luma)

The reverse conversion formula for YUV into RGB is

\[
\begin{align*}
B &= 1.164 \; (Y-16) + 2.018 \; (U-128) \\
G &= 1.164 \; (Y-16) - 0.813 \; (V-128) - 0.391 \; (U-128) \\
R &= 1.164 \; (Y-16) + 1.596 \; (V-128)
\end{align*}
\] (3.3)

3.3.2 Haar Wavelet Transform

The Haar wavelet was proposed by Hungarian mathematician Alfred Haar in 1909 (Hilton 1994) and is the simplest possible wavelet. Nowadays, several definitions of the Haar function and various generalizations as well as some modifications are published and used. The S transform is the integer version of the Harr transform (Calderbank et al 1998) which has the
lowest computational complexity, and reasonably well both for lossy and lossless compression. Consider a sequence of $2^i$ symbols $x(i); x = 1, \ldots, 2^i$. $x(i)$ can be represented as two sequences of symbols, one that represents the low frequency component $l(i)$ (averaging) of $x$ and one that holds the high frequency component $h(i)$ (differencing). The forward S transform equations are given in Equation (3.4) (Park et al 2004).

$$h(i) = x(2i+1) - x(2i)$$

$$l(i) = x(2i) + \left\lfloor \frac{h(i)}{2} \right\rfloor$$  \hspace{1cm} (3.4)

and the reverse transform is,

$$x(2i) = l(i) + h(i) + 1$$

$$x(2i+1) = x(2i) + h(i)$$  \hspace{1cm} (3.5)

where $x(i)$ is the input signal, $h(i)$ is the high frequency sub-band signal and $l(i)$ is the low-frequency sub-band signal. The S transform, which has similarities to the multiresolution Haar representation, can be used for hierarchical methods. Although the S transform is inherently 1-dimensional, by transforming first the rows and then the columns a 2-dimensional transform is achieved. After the rows and columns have both been transformed, one quarter of the image is entirely low frequency components. This band can be transformed recursively to form a multiresolution image pyramid.

3.3.3 Temporal Prediction

Motion estimation obtains the motion information by finding the motion field between the current frame A and the reference frame B. It
exploits temporal redundancy of video sequence, and as a result, the required storage or transmission bandwidth is reduced.

Block matching algorithm is one of the simplest motion estimation techniques that compare one block of the current frame with all of the blocks of the next frame to decide where the matching block is located. Considering the number of computations that has to be done for each motion vector, each frame of the video is partitioned into search windows of size $H*W$ pixels. Each search window is then divided into smaller macro blocks of size $8*8$ pixels. To calculate the motion vectors, each block of the current frame must be compared to all of the blocks of the next frame with in the search range and the Sum of Absolute Difference (SAD) for each matching block is calculated using Equation (3.6) (Iain and Richardson 2003).

$$SAD_{A,B}(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [A(x,y) - B(x+m,y+n)]$$  \hspace{1cm} (3.6)

where $M*N$ is the block size, $A(x,y)$ is the pixel values of current block at $(x+y)^{th}$ position and $B(x+m,y+n)$ is the pixel value of block frame at $(x+m,y+n)^{th}$ position.

![Figure 3.3 Inter frame target window mapping](image)
The block with the minimum value of the Sum of Absolute Difference (SAD) is the preferred matching block. The location of that block is the motion displacement vector for that block in current frame. The best motion vector \((u,v)\) for the target window with the minimum SAD is determined by Equation (3.7) (Ying Li and Khalid Sayood 2007).

\[
(u,v) = \arg \min_{m,n} \{ \text{min SAD}(T_w) \} \tag{3.7}
\]

where \(T_w\) denotes the target window of search range \(H \times W\) as shown in Figure 3.3. If \(b_A(x,y)\) and \(b_B(x,y)\) denote the pixel values of starting location of the specified blocks in current frame block \(A\) and the reference frame block \(B\), respectively and \((u,v)\) indicates the motion displacement of the target block, then the temporal predictor of block \(b_A(x,y)\) can be obtained by,

\[
\hat{b}_A(x,y) = b_B(x + u, y + v) \tag{3.8}
\]

and the temporal prediction residual is

\[
e = b_A(x,y) - \hat{b}_A(x,y) \tag{3.9}
\]

where \(e\) is the prediction residual, \(b_A(x,y)\) is the position of the target block in the current frame and \(\hat{b}_A(x,y)\) is the predicted position of the target block using motion estimation. Equations (3.8) and (3.9) are defined in (Ying Li and Khalid Sayood 2007). The temporally predicted residuals are encoded by the Huffman compression coding scheme. For reconstruction of the frame, the residuals are decoded and added to the predicted pixels.

### 3.3.4 Coding the Residuals

The spatial and temporal prediction residuals are encoded using Huffman codes (Taubman and Marcellin 2001). This coding scheme is an entropy variable length coding. The frequently occurring symbols are
assigned short codes and symbols with less frequency are coded using more bits. The Huffman code can be constructed using a tree. The probability of each intensity level is computed and a column of intensity level with descending probabilities is created. The intensities of this column constitute the levels of Huffman code tree. At each step the two tree nodes having minimal probabilities are connected to form an intermediate node. The probability assigned to this node is the sum of probabilities of the two branches. The procedure is repeated until all branches are used and the probability sum is 1. The code words are constructed by traversing the tree from root to its leaves. At each level 0 is assigned to the top branch and 1 to the bottom branch. This procedure is repeated until all the tree leaves are reached. The codeword for each intensity level consists of 0s and 1s that exist in the path from the root to the specific leaf.

3.4 ALGORITHM FOR ADAPTIVE PREDICTION

Begin

Read image Frame i and Frame i-1 and Correlate
If Correlation Coefficient > 0.99 then
    Apply S wavelet transform to each frame
Divide the frames into macro blocks of equal size m x n
For each block in Frame i, Set search range as –W to + W in Frame i-1
    Map the block of Frame i with respective blocks of Frame i-1
    Calculate minimum abs (difference)
    Motion vector = (x,y) positions of the block whose SAD is minimum
    Calculate MV residual = Actual MV – Predicted MV
    Calculate Frame residual = Actual pixel value – Predicted pixel value
    Encode the error residuals
End for
End
3.5 RESULTS AND DISCUSSION

The proposed algorithm has been implemented using Interactive Data Language (IDL) for standard test video sequences Jane and Tennis of frame size 352x240. From the results, it is obvious that, temporal prediction in wavelet domain has high compression ratio than spatial prediction.

3.5.1 Sample Output

Figure 3.4 shows a snapshot of the execution of the software developed.

![Snapshot of software developed](image)

Figure 3.4 Snapshot of software developed

The prediction results for the Jane sequence frames 1 and 5 are shown in Figures 3.5 and 3.6 respectively. Since frame 0 is the start frame, it is treated as independent I-frame and compressed using JPEG-LS. The frame 1 has high motion changes from frame 0 and hence it is compressed using temporal predictor in spatial domain. Frame 5 has more similarity with Frame 4. Hence its motion estimation is done with wavelet domain.
The prediction results for Tennis sequence frames 1 and 3 are shown in Figures 3.7 and 3.8 respectively. Frame 1 has high motion
characteristics from frame 0 and hence its motion vectors are predicted and residuals are compressed in spatial domain. Frame 3 has more similarity with frame 2 and its motion estimation is done in wavelet domain.

3.5.2 Compression Ratio

To analyze the efficiency of the results of this proposed work, Compression Ratio (CR) and Peak Signal to Noise Ratio (PSNR) parameters are used. Compression Ratio (CR) is defined as the ratio between the number
of bits required to store the image before compression (I) and the number of bits required to store the image after compression (O) as in Equation (3.10) (Hilton 1994).

\[ CR = \frac{O}{I} \]

(3.10)

Table 3.1 lists the motion characteristics of each frame over its previous frame, its prediction mode and compression ratio for the Jane frame sequences (1-9). Frame 0 is the start frame and the adaptive prediction threshold is fixed as 0.99. The frames with correlation coefficient 0.99 and above are considered as having, low motion features and their prediction domain is wavelet. The Frames that have correlation coefficient below 0.99 are high motion frames and hence their prediction is direct spatial domain. Similarly Table 3.2 lists the motion characteristics of each frame over its previous frame, its prediction mode and compression ratio for the Tennis frame sequences (1-9).

**Table 3.1 Compression ratio for test video sequence (Jane frames 1-9)**

<table>
<thead>
<tr>
<th>Frame No.</th>
<th>Original frame size (bits)</th>
<th>Residual frame size (bits)</th>
<th>Correlation coefficient</th>
<th>Prediction domain</th>
<th>Compression ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2027520</td>
<td>325444</td>
<td>0.956559</td>
<td>Spatial</td>
<td>6.23</td>
</tr>
<tr>
<td>2</td>
<td>2027520</td>
<td>262632</td>
<td>0.996758</td>
<td>Wavelet</td>
<td>7.72</td>
</tr>
<tr>
<td>3</td>
<td>2027520</td>
<td>261954</td>
<td>0.996435</td>
<td>Wavelet</td>
<td>7.74</td>
</tr>
<tr>
<td>4</td>
<td>2027520</td>
<td>326492</td>
<td>0.953795</td>
<td>Spatial</td>
<td>6.21</td>
</tr>
<tr>
<td>5</td>
<td>2027520</td>
<td>265730</td>
<td>0.996265</td>
<td>Wavelet</td>
<td>7.63</td>
</tr>
<tr>
<td>6</td>
<td>2027520</td>
<td>323884</td>
<td>0.963063</td>
<td>Spatial</td>
<td>6.26</td>
</tr>
<tr>
<td>7</td>
<td>2027520</td>
<td>261616</td>
<td>0.992512</td>
<td>Wavelet</td>
<td>7.75</td>
</tr>
<tr>
<td>8</td>
<td>2027520</td>
<td>262292</td>
<td>0.991062</td>
<td>Wavelet</td>
<td>7.73</td>
</tr>
<tr>
<td>9</td>
<td>2027520</td>
<td>324404</td>
<td>0.954371</td>
<td>Spatial</td>
<td>6.25</td>
</tr>
</tbody>
</table>
In Table 3.1, frames 1, 4, 6 and 9 are high motion frames as their correlation coefficient is below 0.99. These frames have more motion changes from their reference frames and hence they are predicted in spatial domain although the compression ratio is less than the prediction in wavelet domain. As discussed in Section 3.3.1, motion estimation and prediction in the wavelet domain might be inefficient due to shift invariant properties of wavelet transform. However wavelet domain is excellent for low motion frames due its energy compaction (Gong et al 2004). Hence the other low motion frames 2, 3, 5, 7 and 8 are predicted in wavelet domain and their compression efficiency is high.

Table 3.2 Compression ratio for test video sequence (Tennis frames 1-9)

<table>
<thead>
<tr>
<th>Frame</th>
<th>Original frame size (bits)</th>
<th>Residual frame size (bits)</th>
<th>Correlation coefficient</th>
<th>Prediction domain</th>
<th>Compression ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2027520</td>
<td>295558</td>
<td>0.983613</td>
<td>Spatial</td>
<td>6.86</td>
</tr>
<tr>
<td>2</td>
<td>2027520</td>
<td>301266</td>
<td>0.985742</td>
<td>Spatial</td>
<td>6.73</td>
</tr>
<tr>
<td>3</td>
<td>2027520</td>
<td>256324</td>
<td>0.994431</td>
<td>Wavelet</td>
<td>7.91</td>
</tr>
<tr>
<td>4</td>
<td>2027520</td>
<td>279658</td>
<td>0.997359</td>
<td>Wavelet</td>
<td>7.25</td>
</tr>
<tr>
<td>5</td>
<td>2027520</td>
<td>255678</td>
<td>0.995662</td>
<td>Wavelet</td>
<td>7.93</td>
</tr>
<tr>
<td>6</td>
<td>2027520</td>
<td>254714</td>
<td>0.992310</td>
<td>Wavelet</td>
<td>7.96</td>
</tr>
<tr>
<td>7</td>
<td>2027520</td>
<td>295988</td>
<td>0.971542</td>
<td>Spatial</td>
<td>6.85</td>
</tr>
<tr>
<td>8</td>
<td>2027520</td>
<td>306272</td>
<td>0.981276</td>
<td>Spatial</td>
<td>6.62</td>
</tr>
<tr>
<td>9</td>
<td>2027520</td>
<td>318292</td>
<td>0.975463</td>
<td>Spatial</td>
<td>6.37</td>
</tr>
</tbody>
</table>

In Table 3.2, the Tennis frames 1, 2, 7, 8 and 9 are high motion frames as their correlation coefficient is below 0.99. These frames are predicted in spatial domain although the compression ratio is less than the prediction in wavelet domain. It is obvious that, the temporal prediction in wavelet domain has high compression efficiency than spatial prediction.
However, spatial prediction is faster and motion estimation in the wavelet domain is inefficient for high motion frames.

### 3.5.3 Peak Signal to Noise Ratio

To analyze the quality of the proposed adaptive prediction method, the Peak Signal to Noise Ratio (PSNR) is calculated between the original frame and reconstructed frames by (Hilton 1994),

\[
\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{mse}} \right)
\]  

(3.11)

where, Mean Square Error (mse) is

\[
\text{mse} = \frac{1}{mn} \sum_{x=1}^{m} \sum_{y=1}^{n} (A_{x,y} - B_{x,y})^2
\]  

(3.12)

In Equation (3.12), m and n denote respective number of rows and columns in the image, \( B_{x,y} \) is decompressed image pixel at location (x,y) and \( A_{x,y} \) is original image pixel at location (x,y). Table 3.3 shows the possible combination of spatial-wavelet domains for motion estimation in case of low motion frames and frames with high motion characteristics.

<table>
<thead>
<tr>
<th>Possible Combination</th>
<th>Low motion frame</th>
<th>High motion frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Spatial</td>
<td>Spatial</td>
</tr>
<tr>
<td>C2</td>
<td>Wavelet</td>
<td>Wavelet</td>
</tr>
<tr>
<td>C3</td>
<td>Wavelet</td>
<td>Spatial</td>
</tr>
</tbody>
</table>
Table 3.4 PSNR (dB) value of various samples of Adaptive temporal prediction (Jane sequence)

<table>
<thead>
<tr>
<th>Frame</th>
<th>Motion feature</th>
<th>Spatial</th>
<th>Wavelet</th>
<th>Adaptive</th>
<th>Quality gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>33.75</td>
<td>33.40</td>
<td>33.75</td>
<td>0.35</td>
</tr>
<tr>
<td>2</td>
<td>Low</td>
<td>33.14</td>
<td>33.65</td>
<td>33.65</td>
<td>0.51</td>
</tr>
<tr>
<td>3</td>
<td>Low</td>
<td>32.69</td>
<td>33.91</td>
<td>33.91</td>
<td>1.22</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>32.82</td>
<td>32.54</td>
<td>32.82</td>
<td>0.28</td>
</tr>
<tr>
<td>5</td>
<td>Low</td>
<td>30.57</td>
<td>31.62</td>
<td>31.62</td>
<td>1.05</td>
</tr>
<tr>
<td>6</td>
<td>High</td>
<td>31.65</td>
<td>31.29</td>
<td>31.65</td>
<td>0.36</td>
</tr>
<tr>
<td>7</td>
<td>Low</td>
<td>32.05</td>
<td>33.48</td>
<td>33.48</td>
<td>1.43</td>
</tr>
<tr>
<td>8</td>
<td>Low</td>
<td>31.16</td>
<td>32.64</td>
<td>32.64</td>
<td>1.48</td>
</tr>
<tr>
<td>9</td>
<td>High</td>
<td>33.24</td>
<td>32.21</td>
<td>33.24</td>
<td>1.03</td>
</tr>
</tbody>
</table>

Table 3.5 PSNR (dB) value of various samples of Adaptive temporal prediction (Tennis sequence)

<table>
<thead>
<tr>
<th>Frame</th>
<th>Motion feature</th>
<th>Spatial</th>
<th>Wavelet</th>
<th>Adaptive</th>
<th>Quality Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>High</td>
<td>33.34</td>
<td>32.73</td>
<td>33.34</td>
<td>0.61</td>
</tr>
<tr>
<td>2</td>
<td>High</td>
<td>33.38</td>
<td>32.25</td>
<td>33.38</td>
<td>1.13</td>
</tr>
<tr>
<td>3</td>
<td>Low</td>
<td>32.16</td>
<td>33.85</td>
<td>33.85</td>
<td>1.69</td>
</tr>
<tr>
<td>4</td>
<td>Low</td>
<td>32.08</td>
<td>32.84</td>
<td>32.84</td>
<td>0.76</td>
</tr>
<tr>
<td>5</td>
<td>Low</td>
<td>31.97</td>
<td>33.26</td>
<td>33.26</td>
<td>1.29</td>
</tr>
<tr>
<td>6</td>
<td>Low</td>
<td>32.16</td>
<td>33.12</td>
<td>33.12</td>
<td>0.96</td>
</tr>
<tr>
<td>7</td>
<td>High</td>
<td>33.05</td>
<td>32.34</td>
<td>33.05</td>
<td>0.71</td>
</tr>
<tr>
<td>8</td>
<td>High</td>
<td>33.26</td>
<td>32.72</td>
<td>33.26</td>
<td>0.54</td>
</tr>
<tr>
<td>9</td>
<td>High</td>
<td>33.49</td>
<td>32.81</td>
<td>33.49</td>
<td>0.68</td>
</tr>
</tbody>
</table>
Table 3.4 gives the performance comparison in terms of the quality parameter PSNR for the proposed adaptive method with the existing spatial alone and wavelet alone approaches for Jane sequence. Table 3.5 gives the performance comparison in terms of the quality parameter PSNR for the proposed adaptive method for Tennis sequence. The PSNR values of the adaptive column of various samples of input in Table 3.4 and Table 3.5 show that, the temporal prediction using the proposed adaptive method is robust and hence it can be well adopted for high quality loss less video compression. The efficiency of this adaptive method is working well for any size of frame and the frame size does not affect the performance of this method because the motion estimation is block based. Each frame is divided into blocks of equal size and then the matching and prediction is made. However as the frame size increases the matching time will increase, but the domain selection and the compression quality remain unchanged. The compression quality of reconstructed frames obtained by this adaptive method is higher than that of the results of some earlier methods (Xiang et al 1999, Ming-Feng et al 2003, Park et al 2004) found in the literature. As this adaptive prediction method is block based, the compression speed of this method is faster than the pixel based method proposed by Ying Li and Khalid Sayood (2007).