CHAPTER 3

WASH TREE ALGORITHM BASED VIDEO CODING

3.1 INTRODUCTION

SVC is a promising technology for spatial, temporal and quality scalability on a heterogeneous or a home network environment. However, SVC has various limitations such as coding performance and complex decoding. SPIHT algorithm based on wavelet transform evolved as the strong candidate to overcome the foresaid setbacks (Said and Pearlman 1996). However, SPIHT algorithm based coding technique requires large amount of time and memory to accurately accomplish the job, which causes delay in real time video transmission. Eventually, it requires large amount of bandwidth. WASH Tree based video coding algorithm with low memory is proposed in this chapter. The performance of this video codec is evaluated using different spatial and temporal scalable parameters like, good image and video quality with high PSNR; fast coding and decoding and a fully progressive bit-stream ability to code for exact bit rate (Jun Ho Cho et al 2009).

3.2 WASH TREE ALGORITHM BASED VIDEO CODING

A WASH Tree algorithm is known as weighted adaptive scalable hierarchical tree. Video coding based on WASH Tree algorithm is proposed to improve the image and video quality with reduced processing, along with less memory consumption. A WASH Tree algorithm is the enhanced version of SPIHT algorithm associated with adaptive scalable operator to avoid
multiple complex sorting and refinement passes. The new prescalar is derived from general arithmetic filters (Sahoo and Szekelyhidi 2001). The coding and decoding of the WASH Tree algorithm are shown in Figures 3.1 and 3.2 respectively. The information source consists of different video test sequences with various resolution ranges.

Figure 3.1 Block diagram of WASH tree algorithm based encoder

Figure 3.2 Block diagram of WASH tree algorithm based decoder
3.2.1 WASH Tree Coder

The concepts involved in WASH Tree coding are explained below:

The encoding of video signals involves the following major steps:

i. Initialization
ii. Prescaling
iii. Sorting pass
iv. Refinement pass
v. Quantization step update

To begin with, the video sequence is converted as number of frames depending on its rate and the extracted image sequences are passed through prescaling operators. Scalar quantization and prescaling operations are implemented by division and/or multiplication by constant parameters (controlled by a quantization parameter or quantizer step size). In general, gain is achieved through quantization integration and prescaling multiplications. In this work, two simple scalable operators are proposed as given in equations 3.1 and 3.2, which make effective changes with each and every frame based on its density.

\[
\hat{f}(i, j) = f(i, j) - 0.25 \times \begin{cases} f(i + t, j + t) - f(i - t, j - t) \\ -f(i + t, j - t) - f(i - t, j + t) \end{cases} \tag{3.1}
\]

\[
\hat{f}(i, j) = f(i, j) - 0.25 \times \begin{cases} f(i + t, j) - f(i, j + t) \\ -f(i, j - t) - f(i - t, j) \end{cases} \tag{3.2}
\]

where \( f(i, j) \) is the image function

\( \hat{f}(i, j) \) is the modified image function
$i, j$ are the row column vector coordinates

t is the time space integer variable

For example, original test sequence and its prescaled response are shown in Figure 3.3(a) and Figure 3.3(b).

**Figure 3.3(a) Original test sequence

Figure 3.3(b) Prescaled sequence**
Discrete wavelet coding algorithm (Karlekar and Desai 1996) (Shapiro 1993), views wavelet coefficients as a collection of spatial orientation trees, with each tree consisting of coefficients from all sub-bands that correspond to the same spatial location in an image. Normally, most of the image energy in DWT is concentrated in the low frequency components (shown with * in Figure 3.4). The variance decreases from highest to the lowest (coarsest to finest) levels of subband pyramid. A tree structure, called spatial orientation tree, defines the spatial relationship on the hierarchical pyramid, similar to zerotree structure of embedded zerotree wavelet (EZW) (Xun Guo et al 2006) (Kim, Xiong and Pearlman 2000) (Kim and Pearlman 1997). Figure 3.4 shows the spatial orientation tree defined in a pyramid constructed with recursive four-subband splitting. Each node of the tree corresponds to a pixel, and is identified by pixel coordinate. Its direct descendants (offsprings) correspond to 2 x 2 adjacent pixels of same spatial orientation in the next finer level of the pyramid. The tree is defined in such a way that each node has either no offsprings (leaves) or four offsprings as shown in Figure 3.4. The pixels in the highest level of the pyramid are the tree roots (Ekram Khan and Mohammed Ghanbari 2002) (Danyali and Mertins 2002) (Yong Sun et al 2002).

Initially, it uses multi-pass zero-tree coding to transmit large wavelet coefficients (in magnitude) (Ali Aghagolzadeh et al 2008) (Li Wern Chew et al 2008). WASH tree achieves embedded coding in the wavelet domain using three lists: list of significant pixels (LSP), list of insignificant pixels (LIP), and list of insignificant sets (LIS). A set of tree coefficients are significant if the largest coefficient magnitude in the set is greater than or equal to a certain threshold (e.g., a power of two), else, it is insignificant. In each pass, the significance of a larger set in the tree is tested and if the set is insignificant, a binary zero-tree bit is used to set all coefficients in the set to zero; otherwise, the set is partitioned into subsets for further significance tests.
as shown in Figure 3.4. The threshold is halved before the next pass if all the coefficients are tested in first pass. The underlying assumption of WASH tree coding is that most images can be modeled as having decaying power spectral densities. When the thresholds are powers of two, WASH tree (Sudhakar, Karthiga and Jayaraman 2005) (Wen-Chien Yen and Yen-Yu Chen 2005) (Shaorong Chang and Lawrence Carin 2006) coding resembles a bit-plane coding scheme, which encodes one bit-plane at a time (starting from the most significant bit) and flow diagram of test pass is shown in Figure 3.5. The WASH tree coder performs competitively with most other coders (Kim, Xiong, and Pearlman 2000) (Pearlman, Kim and Xiong 1998) (Wang et al 1999).

![Figure 3.4 Spatial orientation of tree diagram](image)
Algorithm:

The proposed WASH Tree algorithm for encoding is given below:

Step 1: Initialization

Step 2: Prescale operation

\[
\hat{f}(i, j) = f(i, j) - 0.25 \times \begin{cases} f(i+t, j+t) - f(i-t, j-t) \\ -f(i+t, j-t) - f(i-t, j+t) \end{cases}
\]

Output set the LSP as an empty list, add the coordinates to the LIP, add the coordinates with descendants to the list LIS, as type A entries,
Step 3: Sorting Pass

(i) for each entry \((i, j)\) in the LIP do: output \(S_n(i, j)\), if \(S_n(i, j) = 1\) move \((i, j)\) to the LSP, output the sign of \(C_{i, j}\).

(ii) for each entry \((i, j)\) in the LIS do:

(iii) if the entry is of type A then output \(S_n(D(i, j))\), if \(S_n(D(i, j)) = 1\) then * for each \((k, l) \in o(k, l)\) do: output \(S_n(k, l)\),

if \(S_n(k, l) = 1\) then add \((k, l)\) to the LSP, output the sign of \(C_{k, l}\),

if \(S_n(k, l) = 0\) then add \((k, l)\) to the end of the LIP, *if \(L(i, j) \neq 0\) then move \((i, j)\) to the end of the LIS, as an entry of type B, go to Step (iv).

Otherwise remove entry \((i, j)\) from the LIS,

(iv) if the entry is of type B, output \(S_n(L(i, j))\),

if \(S_n(L(i, j)) = 1\) then * add each \((k, l) \in o(i, j)\) to the end of the LIS as an entry of type A, * remove \((i, j)\) from the LIS,

Step 4: Refinement Pass

For each entry \((i, j)\) in the LSP, except those included in the last sorting pass (i.e., with the same \(n\)), output the \(n\)th most significant bit of \(|C_{i, j}|\),
Step 5: Quantization-Step Update- Decrement \( n \) by 1 and go to Step 2.

NOTE: Since decoding process is symmetric about encoding process.

3.3 SIMULATION RESULTS AND DISCUSSION

Based on the experiments conducted to study the performances of WASH tree algorithm, different kinds of benchmarks have been identified. Summarized experimental results with necessary parametric analysis and comments are revealed as follows. Simulation is performed using MATLAB with the parameters provided in Table 3.1. Also, graphical user interface (GUI) is built to make it easier to monitor the impact of changing parameters, such as frame size, compression ratio, PSNR, encode time and decode time as the program is run. Three different video test sequences are employed to examine and analyze the proposed WASH tree algorithm as shown in Table 3.1.

Table 3.1 Table of test sequences

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Sequences</th>
<th>Resolution</th>
<th>Number of frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mother and daughter</td>
<td>176x144</td>
<td>96</td>
</tr>
<tr>
<td>2</td>
<td>Hall monitor</td>
<td>176x144</td>
<td>96</td>
</tr>
<tr>
<td>3</td>
<td>News reader</td>
<td>176x144</td>
<td>96</td>
</tr>
</tbody>
</table>

To realize the performance parameter of WASH tree algorithm, the developed GUI is shown in Figure 3.6(a). News reader video sequence is browsed as an input file and number of frames separated from input video sequence is provided in Figure 3.6(b). A total of 96 frames are considered at a rate of 10 frames per second (fps) with frame resolution 176 x 144. Figures 3.6(c), 3.6(d) and 3.6(e) portrait the encoding process, decoding process and validated results with PSNR, compression ratio, encode time and decode time respectively.
Figure 3.6(a) GUI for video compression system (News reader)

Figure 3.6(b) Frame Separation
Figure 3.6(c) Encoding Process

Figure 3.6(d) Decoding Process
In addition, mother and daughter video sequence with frame resolution 176 x 144 is considered for performance evaluation and is illustrated in Figure 3.7. Figure 3.7(a) depicts GUI for video compression system and the original mother and daughter video sequence is browsed as an input sequence. Number of frames separated from input video sequence is provided in Figure 3.7(b). A total of 96 frames are considered at a rate of 10 frames per second with frame resolution 176 x 144. Figures 3.7(c), 3.7(d) and 3.7(e) describes the encoding process, decoding process and validated results with PSNR, compression ratio, encode time and decode time respectively. Furthermore, original hall monitor sequence is considered for parameters observation is given in Figure 3.8(a). Similar types of validation is made to verify the robustness of the proposed WASH tree algorithm introduced in Figure 3.8(b), 3.8(c), 3.8(d) and 3.8(e).
Figure 3.7(a) GUI for video compression system (Mother and daughter)

Figure 3.7(b) Frame Separation
Figure 3.7(c) Encoding Process

Figure 3.7(d) Decoding Process
Figure 3.7(e) Validated results

Figure 3.8(a) GUI for video compression system (Hall monitor)
Figure 3.8(b) Frame Separation

Figure 3.8(c) Encoding Process
Figure 3.8(d) Decoding Process

Figure 3.8(e) Validated results
Figure 3.9(a) shows PSNR value in dB with respect to corresponding frame for LM SPIHT and WASH tree algorithms. The result conveys that the PSNR value of the mother and daughter video sequence obtained through WASH tree algorithm outperforms LM SPIHT algorithm. Evaluation is also performed for hall monitor video sequence with PSNR versus frame and the result is illustrated in Figure 3.9(b). Similar type of performance observed in Figure 3.9(c) for news reader sequence. This performance improvement with PSNR value is due to significant prescale test formula which scales down the dynamic range of spatial resolution. Also, unnecessary lists are discarded and the process length of sorting phase is shortened to reduce the coding time and memory usage (Liang Zhang, Demin Wang and Andre Vincent 2008).

![Figure 3.9](image_url)  
**Figure 3.9**  (a) Frame number versus PSNR for Mother and daughter sequence
Figure 3.9(b) Frame number versus PSNR for Hall monitor sequence

Figure 3.9(c) Frame number versus PSNR for News Reader sequence
Average PSNR values obtained for various video sequences with different coding algorithms are compared and tabulated in Table 3.2 and the same is projected as 3D cylindrical chart which is shown in Figure 3.10. Also, the attained average compression ratio is 60.46306. The proposed WASH Tree algorithm saves processing time and provides stable quality results as demonstrated in the time (seconds) columns of the Table 3.3 (Shang-Hsiu Tseng and Aldo Morales 2009). The proposed WASH Tree algorithm produces smaller improvement in PSNR as compared to 3D SPIHT and LM SPIHT shown in Table 3.2. Further, it is significant to maintain good quality and also consumes less processing time and very good compression ratio.

**Table 3.2 PSNR Comparison for various algorithms**

<table>
<thead>
<tr>
<th>Sequences</th>
<th>3D SPIHT</th>
<th>LM SPIHT</th>
<th>WASH Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mother and daughter</td>
<td>34.62</td>
<td>34.87</td>
<td>35.13</td>
</tr>
<tr>
<td>Hall monitor</td>
<td>31.97</td>
<td>32.94</td>
<td>33.14</td>
</tr>
<tr>
<td>News reader</td>
<td>34.39</td>
<td>34.57</td>
<td>35.0243</td>
</tr>
</tbody>
</table>

**Table 3.3 Processing time comparison for various algorithms**

<table>
<thead>
<tr>
<th>Sequences</th>
<th>3D SPIHT</th>
<th>LM SPIHT</th>
<th>WASH Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Encode Time (s)</td>
<td>Decode Time (s)</td>
<td>Encode Time (s)</td>
</tr>
<tr>
<td>Mother and daughter</td>
<td>56.37</td>
<td>14.67</td>
<td>34.18</td>
</tr>
<tr>
<td>Hall monitor</td>
<td>59.44</td>
<td>23.81</td>
<td>46.68</td>
</tr>
</tbody>
</table>
3.4 SUMMARY

WASH tree algorithm is proposed for video compression technique to improve coding efficiency in this chapter. Three different video test sequences are considered with the same specifications of frame rate is 10fps, resolution 176x144 and number of frame is 96. The robustness of the proposed algorithm is measured by PSNR and compared with existing techniques. The increased PSNR obtained in simulation result shows that the proposed technique achieves significant improvement in performance. The performance improvement with PSNR value is due to spatially significant scale down preprocessing technique which considerably reduces the dynamic range of spatial resolution.