Abstract – Rate scalable video compression techniques are very attractive, because they are able to encode the video in embedded way that makes the decoding process more flexible to the bandwidth changes. Embedded wavelet coding technology plays an important role in this area. Traditional wavelet transforms are globally optimized. It is very efficient to code the natural images where sudden transitions rarely exist. On the other hand, video compression mostly deals with the differences between adjacent frames, also known as delta frames. Unfortunately, commonly used wavelet transforms are not so efficient on delta frame coding. In this paper, we are presenting a new wavelet coding technique – Flexible Block Wavelet Coding (FBWC). It is specially designed for delta frame compression. FBWC not only makes delta frame compression more efficient but also keeps the rate scalability of wavelet based rate scalable video compression techniques, and other scalabilities as well. Based on this new delta frame compression technique, we have implemented a highly efficient rate scalable video codec. Its overall performance surpasses the video codec based on SPIHT and is comparable to that of H263.

INTRODUCTION

Wavelet based rate scalable video compression techniques are very popular recently, because of their scalabilities in data rate, spatial and PSNR. It also draws a lot of attention from research community. Shapiro [1] has laid the foundation for rate scalable still image compression by introducing Embedded Zerotree Wavelet (EZW). Said [2] presented a new and different implementation based on Set Partitioning in Hierarchical Trees (SPIHT). Said’s method provides better performance in compression and speed. The natural extension of the embedded coding methods is to apply them in video compression. Several research groups have adapted the EZW into video compression domain. The major challenges of
those adaptations come from two aspects: First is how to explore the temporal redundancy among adjacent frames. Second is how to compress the Predicted Error Frame (PEF) using EZW or its variants.

Several approaches have been proposed to explore the temporal redundancy among adjacent frames. S. A. Martucci [3] used the Block Motion Estimation (BME) from H.263 followed by an Overlapped Block Motion Compensation (OBMC) to get PEF. The purpose of using OBMC is to reduce the boundaries between 16X16 blocks. These boundaries are introduced by BME and very difficult for traditional wavelet methods to compress, because they usually appear as sharp changes. H. Klock [4] takes another approach. After the BME from H.263, He uses Region-based Motion Compensation (RMC) based on Chroma Key technique. Kim, Xiong and Pearlman [5] implemented BME in hierarchically wavelet domain to reduce the complexity of full search. Notice that all the methods we discussed so far are based on some kind of BME. It does effectively and efficiently reduce the temporal redundancy, but it also introduces the boundary discontinuities along block edges. In addition, Inter blocks (difference blocks) and intra blocks (independent blocks) may be next to each other. Because of their different nature, the boundary problem can be even worse.

To address the PEF compression issue, basically there are two ways to tackle this problem. Use 2-D EZW or its variants to compress the PEFs as still images, such as Martucci’s Zerotree Entropy (ZTE) coding [3]. The other way is using 3-D EZW or SPIHT to compress Group of PEFs. The most commonly used the wavelet for temporal direction is Haar wavelet. The examples are Klock’s spatial-temporal (3D) wavelet transform [4] and Pearlman’s 3-D SPIHT [5].

Since the dominant motion estimation and compensation techniques for video compression are block based, the inherent boundary discontinuities along block edges still exist, although all kinds of smoothing techniques have been used. In fact, PEFs are different than the natural images by their definition. Therefore, we think that PEFs should be compressed differently than natural images.

In this paper we describe a new Flexible Block Wavelet Coding (FBWC) technique. It is specially designed for PEF compression. It surpasses the SPIHT in compressing PEFs. A rate scalable video codec has been implemented using FBWC for PEF compression.

**OVERVIEW OF THE ENCODER**

This section gives an overview of our encoder. The system block diagram is shown in Figure 1.

When encoder gets an input frame, our scene change detector will tell encoder the current input frame should be compressed either as independent frame (key frame) or delta frame except the first frame that has to be compressed as key frame. The key frames are compressed using SPHIT. The embedded feature of SPHIT makes it possible to encode the key frame at exactly the chosen rate. For maintain relative constant quality cross broad, we allocate 3 times budget for key frame than that for delta frame.
The delta frames are compressed in four steps: First a BME from H.263 is used to detect the local motion against the reconstructed previous frame. In our system it is done to half-pixel accuracy on block of 16X16. Next, each block is predicted using the overlapping block motion compensation scheme. After all the blocks have been predicted, the residuals are pieced together to form a complete PEF for subsequent processing and the motion vectors will be coded using Variable Length Coding (VLC) from H.263. Thirdly, The PEFs will be transformed into wavelet domain using Flexible Block Wavelet Transform (FBWT). Finally, zerotree coding mechanism and entropy coder will be used to actually compress the FBWT transformed wavelet coefficients. We will discuss the technical details in next section.

![Figure 1: Block Diagram of Encoder.](image)

**TECHNICAL DETAILS OF FBWC**

**Color Transform and Bit Allocation Among Color Bands**

Since our video codec is designed for color video compression, the most popular color format around is RGB color. As we all know, Red, Green and Blue are highly correlated, so that it is necessary to convert the RGB color into a new color space which has less correlation among color bands. We have chosen YUV color space. Efficient bit allocation among Y, U and V bands is a very challenge topic. Said [2] introduced CSPIHT based on Karhunen-Loeve (KL) transform. K. Shen [6] proposed a CEZW based on Space Oriented Tree (SOT). We have developed an efficient and effective bit allocation scheme for YUV color bands that matches or surpasses CEZW in terms of PSNR and is much fast than CSPIHT. We first down sample U and V bands and keep Y band the same, then zerotree coding method will
be applied to Y, U and V bands alternatively for each threshold until the target maximum data rate is reached or the threshold becomes less than 1.

**Motion Estimation and Compensation**

The proposed codec uses the BME and OBMC techniques of H.263.

**Flexible Block Wavelet Transform (FBWT)**

In order to deal with the motion compensation boundary discontinuity problem, we have specially designed a set of biorthogonal wavelet transforms with boundary treatments. For each 16X16 (8X8) block in a PEF, we choose the most suitable wavelet transform from our candidate set according to the block size and the transform level. The picked wavelet transform will be used to do the current level of transform. Let us use 8X8 block PEF as an example. After FBWT, for an 8x8 block, the coefficients look like:

\[
\begin{bmatrix}
L^1_{1x1} & HD^1_{1x1} & HD^2_{1x2} & HD^4_{4x4} \\
VD^1_{1x1} & DD^1_{1x1} & HD^2_{2x2} & HD^4_{4x4} \\
VD^2_{2x2} & DD^2_{2x2} & HD^4_{4x4} & HD^4_{4x4} \\
VD^4_{4x4} & DD^4_{4x4} & HD^4_{4x4} & HD^4_{4x4}
\end{bmatrix}
\]

Where, \(HD^2_{2x2}\) represents 2x2 coefficients for horizontal high frequency at second level transform, etc. Then, we rearrange those coefficients to their corresponding places as following:

```
<table>
<thead>
<tr>
<th>L^3</th>
<th>HD^3</th>
<th>HD^2</th>
<th>HD^1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VD^3</td>
<td>DD^3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VD^2</td>
<td>DD^2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VD^1</td>
<td>DD^1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Here, \(HD^2\) includes all coefficients for horizontal high frequency at second level in all of the 8x8 blocks. The rearranged wavelet coefficients are ready for embedded coding. By using FBWT, we can handle the motion compensation boundary gracefully. Meanwhile all the scalabilities of embedded wavelet coding are still preserved.
Rate Scalability and Reference Frame

As mentioned by Shen and Delp [6], to make a video stream truly rate scalable, we need to keep the reference frames the same at both the encoder and the decoder. This can be achieved by adding a feedback loop in the decoder, such that the decoded reference frames at both the encoder and decoder are locked to the same data rate -- the lowest data rate. We assume that the target data rate $R$ is within the range $R_L \leq R \leq R_H$ and the bits required to encode the motion vector fields have data rate $R_{MV}$, where $R_{MV} < R_L$. At the encoder, since $R_{MV}$ is known, the embedded bit stream can always be decoded at rate $R_L - R_{MV}$, which is then added to the predicted frame to generate the reference frame $P_{ref}$. At the decoder, the embedded bit stream is decoded at two data rates, the targeted data rate $R - R_{MV}$ and the fixed data rate $R_L - R_{MV}$. The frame decoded at rate $R_L - R_{MV}$ is added to the predicted frame to generate the reference frame, which is exactly the same as the reference frame $P_{ref}$ used in the encoder. The frame decoded at rate $R - R_{MV}$ is added to the predicted frame to generate the final decoded frame. This way, the reference frames at the encoder and the decoder are identical, which leaves the decoded PEF $P_{diff}$ as the only source of distortion. Hence, error propagation is eliminated.

EXPERIMENTAL RESULTS AND ANALYSIS

Figure 2 shows the performance comparison chart of FBWC based codec, normal SPIHT based codec and video compression standard H263 on carphone sequence. Carphone sequence has frame size 176X144, 10 fps (by coding every third frame) and was compressed at 30 kbps. H263 was used with all the advanced options (Annex D, F and E). The overall PSNRs of each frame in the sequences are shown. The average PSNRs for FBWC, H263 and SPIHT over total 127 frames are 30.07, 30.67 and 28.91 respectively. Since FBWC and SPIHT are using exactly the same compression methods except the wavelet transforms, it means that FBWT contributes 1.16db quality improvement by itself. There is still 0.6db difference between FBWC and H263, but notice that the major difference occurs in the first half of the clip where the motion is little. In fact, FBWC is comparable with H263 in medium and fast motion scenarios. Overall FBWC performs much better than traditional wavelet based video compression and comparably with H263.
References


