Side information generation in Wyner-Ziv decoder

Abstract

Wyner-Ziv (WZ) encoder is a low complexity encoder and can be made to achieve compression comparable to traditional high complexity encoders but at the expense of a high-complexity decoder. The high complexity of the decoder is mainly attributed to the generation of side information which involves motion estimation. In this paper sub-pixel motion estimation and intra-prediction based on H.264 standard are considered for side information generation. This method is tested with different test sequences by measuring quality of the decoder output video and comparing with other simpler methods. The H.264 standard based side information generation is found to be superior over other methods considered.

Key Words: Wyner-Ziv coding, distributed video coding, side information generation, sub-pixel motion estimation, intra prediction

1. Introduction

Traditional video coding involves complex encoding and relatively simple decoding. The complexity of the encoder is mainly due to motion estimation[3] Wyner-Ziv video coding has reverse complexity with encoder being less complex compared to decoder[9]. In traditional video codecs the spatial and temporal correlation in video are exploited at the encoder side via prediction using transforms, motion estimation, entropy coding etc. In the Wyner-Ziv coding this is done at the decoder side and hence decoder is more complex compared to the encoder.

There is lot of work done on implementing Wyner-Ziv codec using error control codes like turbo codes[2] and LDPC codes[5][10][12]. The error control codes are required to correct the decoder side prediction error. This prediction is called side information (SI) generation and several motion estimation (ME) schemes have been tested for SI generation[11]. The standard based SI generation is also considered[1]. In this previous research[1] LDPC codes for error control and H.263 standard based motion estimation and intra prediction for SI generation are used. In our approach, we consider LDPC codes for error control and H.264 standard based motion estimation and intra prediction for SI generation. The basic Wyner-Ziv encoder and decoder are implemented to test H.264 standard based SI generation method. This is then compared against other SI generation methods.

2. WZ Encoder

The WZ encoding process involves encoding of key frames and WZ frames[4]. The overall encoding process is illustrated in Figure 1. Some of the input frames are marked as key frames and encoded using H.264 Intra frame[6][7][14][8] encoding. The WZ frames are encoded as follows: The difference between previous reconstructed key frame and WZ frame is quantized using a uniform scalar quantizer and the output is encoded using low-density-parity-check accumulated (LDPCA) code.

![Figure 1. Block diagram of WZ encoder[4]](image)

3. WZ Decoder

The first step in decoding a WZ frame is generation of side information (SI) using key frames[4]. The SI frame generated is used by the low density parity check accumulate (LDPCA) decoder to decode the WZ bit-stream and generate the WZ frame. The previous key frame is subtracted from the SI frame generated to produce error frame which is subsequently quantized. This is used by LDPC decoder to correct bit errors in WZ encoded error.
frame. The error frame obtained in this way is added to the key frame and de-quantized to reconstruct the WZ frame. At low bitrates some of the macroblocks in the WZ frame cannot be reconstructed. These macroblocks are replaced with the corresponding macroblocks from WZ estimated frame.

4. Generation of Side Information (SI)

The SI frame generation is key aspect of WZ decoding process. The quality of the decoding is dependent on the SI frame and in terms of complexity this is a major component of WZ decoder.

The generation of SI is illustrated in Figure 3 and it involves,

A. Motion estimation (ME) between two key frames to obtain motion vectors (MV). The estimation is done in both forward and backward directions to obtain $MV_f$ and $MV_b$ respectively as shown in Figure 3. The block sizes are used for ME are 16x16, 8x8 and 4x4.

B. The derivation of motion vectors for WZ frames. This is done by scaling MVs obtained in the previous step by the ratio calculated as distance between WZ frame to previous key frame to the distance between key frames themselves. In Figure 3, the scaling factor is $\frac{1}{2}$ since the ratio of distance between key frame and WZ frame to distance between two key frames is $\frac{1}{2}$.

C. Obtaining the estimation for macroblock of a WZ frame by interpolation of macroblocks from the previous and next key frames. The motion vectors calculated in the previous step are used here to obtain mapping of macroblocks in WZ frame to key frame macroblocks. The forward predicted frame ($PF$) is obtained using forward motion vector $MV_f$ and backward predicted frame ($PB$) is obtained using backward motion vector $MV_b$.

Then the side information frame $Y$ is obtained as $(PF + PB)/2$ as shown in Figure 3.

5. A. Forward and Backward Motion Estimation

The motion estimation for future key frame is done with respect to previous key frame followed by vice versa operation yielding two motion vectors namely forward (MVF) and backward (MVB) prediction MVs for a 4x4 block. It is not necessary that the previous and future frames have to be key frames but any available decoded frames can be used for this purpose. For example consider the case where key frame distance is four then the decoding of frames is done as shown in the Figure 4. Here S2 is
decoded first using side information $S_{2\text{st}}$ generated using reconstructed key frames $K_{1R}$ and $K_{2R}$. This is followed by $S_1$ for which $K_{1R}$ and $S_{2R}$ are used for generating side information $S_{1st}$. Finally $S_3$ is decoded for which side information $S_{3st}$ is generated using $S_{2R}$ and $K_{2R}$ reconstructed frames. The forward and backward motion vectors of the current frame are obtained by appropriately scaling and changing the signs of forward and backward prediction MVs obtained earlier.

The ME for WZ decoding is based on least sum of absolute difference (SAD) and not on the total cost which is sum of residual and MV bits. But for encoding ME based on cost is required. It is desirable to have single ME which produces two MVs - one based on least SAD and the other based on least cost. Hence the proper modifications need to be done for existing ME to have two sets of contexts and update them as the search progresses. Similar modifications need to be done even for early exit criterion so that both the set of MVs obtained are near optimal.

For the case of key frame distance shown in the Figure 4, the operations for generating MVs can be summarized as,

1. **Obtaining MVs for $S_2$:** Perform ME between $K_2$(frame under consideration) and $K_1$ (reference) to obtain a set of forward MV based on SAD. Similarly perform ME between $K_1$(frame under consideration) and $K_2$ (reference) to obtain a set of backward MV. Now for each block in $K_2$ with a given forward MV calculate the position of same block in $S_2$. This position is calculated by shifting the position of block in $K_2$ by $\frac{1}{2}$ * forward MV. Also the forward MV for this block in $S_2$ will be $\frac{1}{2}$ * forward MV. Similarly for each block in $K_1$ with a given backward MV calculate the position of same block in $S_2$. This position is calculated by shifting the position of block in $K_1$ by $\frac{1}{2}$ * backward MV. Also the backward MV for this block in $S_2$ will be $\frac{1}{2}$ * backward MV.

2. **Obtaining MVs for $S_1$:** Operations involved in the calculation of forward and backward MV for $S_1$ are similar to those of $S_2$ with the exceptions that prediction MVs are generated using $K_1$ and $S_2$.

3. **Obtaining MVs for $S_3$:** Here prediction MVs are generated using $S_2$ and $K_2$.

6. **Sub-pixel motion estimation for $S_1$ generation**

The side information generated can be improved by using sub-pixel motion vectors for both forward and backward predictions. In order to derive these sub-pixel positions interpolation between pixels needs to be performed. For half pixel motion estimation there are three pixel positions that need to be evaluated. For quarter pixel motion estimation there are twelve pixel positions that need to be evaluated. The generation of sub-pixel positions is done as per H.264 standard[6][7][14][8] and is briefly described below.

6.1. **Half-pixel positions**

In Figure 5 the pixel positions numbered H33, G33 and D33 are half pixel positions and need to be derived.

These are generated by interpolating full pixel or half pixel values using a six tap filter [1 -5 20 -5 1]/32. Following equations can be used,

- $H_{33} = \left[ F_{13} - 5 * F_{23} + 20 * F_{33} + 20 * F_{43} - 5 * F_{53} + F_{63} + 15 \right] >> 5$
- $G_{33} = \left[ F_{31} - 5 * F_{32} + 20 * F_{33} + 20 * F_{34} - 5 * F_{35} + F_{36} + 15 \right] >> 5$
- $D_{33} = \left[ H_{31} - 5 * H_{32} + 20 * H_{33} + 20 * H_{34} - 5 * H_{35} + H_{36} + 15 \right] >> 5$

6.2. **Quarter-pixel positions**

The quarter pixels are obtained by averaging nearest full pixel or half pixel positions.
The following equations are used for obtaining quarter pixel positions,

- \( q_1 = (F33 + G33 + 1) >> 1 \)
- \( q_2 = (G33 + F34 + 1) >> 1 \)
- \( q_3 = (F33 + H33 + 1) >> 1 \)
- \( q_4 = (H33 + G33 + 1) >> 1 \)
- \( q_5 = (G33 + D33 + 1) >> 1 \)
- \( q_6 = (G33 + H34 + 1) >> 1 \)
- \( q_7 = (H33 + D33 + 1) >> 1 \)
- \( q_8 = (D33 + H34 + 1) >> 1 \)
- \( q_9 = (H33 + F43 + 1) >> 1 \)
- \( q_{10} = (H33 + G43 + 1) >> 1 \)
- \( q_{11} = (D33 + G43 + 1) >> 1 \)
- \( q_{12} = (G43 + H34 + 1) >> 1 \)

The forward and backward predicted data obtained for each partition block is averaged to obtain the final prediction block. In case for a block if there is no motion vector, then intra prediction can be used to predict the block from neighboring pixels. The improvement in the quality of SI generated with the sub pixel motion estimation over full pixel motion estimation can be measured both visually and quantitatively[15]. The quantitative measurement can be done by PSNR of the predicted frame with reference to the original frame. The objective is to get a good improvement in the quality of SI frame.

7. Motion Compensation (MC)
Motion compensation involves obtaining the forward and backward prediction blocks for all the 4x4 blocks in current frame using the motion vectors available for that block. Depending on the motion vectors available for a block there are four possibilities of MC.
1. **FORW_MV** is available: Forward motion compensation is done to obtain forward predicted frame (PF) as shown in the Figure 3.
2. **BACK_MV** is available: Backward motion compensation is done to obtain backward predicted frame (PB) as shown in the Figure 3.
3. **FORW_MV and BACK_MV** are available: Both forward and backward motion compensations are done.
4. **No MV exists**: No motion compensation is done for the block. This block is candidate for intra prediction which is discussed in section 8. Interpolation filters and the steps used to obtain the prediction block for sub-pel MVs are same as those defined in H.264 standard [6][7][14][8].

8. Bidirectional Interpolation
The final prediction for a 4x4 block with two motion vectors is obtained by the interpolation of forward and backward prediction blocks obtained in previous step. The method used here is simple average with rounding.

9. Intra prediction
There are two cases where Intra prediction is used for a block,
1. Blocks with no motion vectors:
2. Blocks where the SAD value for the predicted MV is high:

   In order to determine whether SAD value is high or not an adaptive threshold based on quantizer is used for comparison. Intra prediction is done as defined in H.264 standard[6][7][14][8] using the neighboring blocks. There are nine different directional prediction modes as shown in the Figure 7 and they are dependent on the availability of left, top, top left and top right neighbors. The availability condition is modified here and only the case of blocks at the frame boundaries where some of the neighboring blocks are not available is considered.

![Figure 7. Nine spatial prediction modes used in H.264 standard](image)

![Figure 8. 4x4 block Intra prediction](image)
The 4x4 block intra prediction for a macroblock uses left, top, top left and top right 4x4 blocks as shown in the Figure 8. There are 16 blocks namely a, b...p and for each 4x4 block a prediction block is generated using neighboring blocks. For example

- Block 'a' has 'I' as left neighbor, 'A' as top neighbor, 'M' as top left neighbor and 'B' as top right neighbor.
- Block 'e' has 'J' as left neighbor, 'a' as top neighbor, 'I' as top left neighbor and 'b' as top right neighbor.
- Block 'b' has 'a' as left neighbor, 'B' as top neighbor, 'A' as top left neighbor and 'C' as top right neighbor.
- Block 'f' has 'e' as left neighbor, 'b' as top neighbor, 'a' as top left neighbor and 'c' as top right neighbor.

The prediction block is generated using one of the direction modes shown in Figure 7. The direction mode to be used for the block can be derived from the direction mode of the co located block in the previous key frame. The prediction block generated is a linear combination of neighboring blocks and the linear combination depends on the direction mode.

10. Results
Currently three different schemes are implemented for WZ frame generation using JM reference software[13].
1. Simple averaging of two key frames
2. Full pixel motion estimation,
3. Half-pixel motion estimation
The even frames are encoded as I frames and odd frames are encoded using WZ encoder[4]. For quality comparison between WZ encoder and H.264 encoder a separate encoding is done with even frame being I frame and odd frame being P frame. The WZ frame obtained using SI prediction is analyzed for PSNR with reference to corresponding H.264 P frame. The reconstructed frames using different SI generation schemes is shown in Figure 9. The quality of reconstruction frames in case of motion estimation schemes is clearly superior to simple averaging scheme which can be observed in Figure 9. The average PSNR plot for a QCIF (176x144) test sequence for a ME search range of 64 is shown in Table 1 and the plot is shown in Figure 10.

![Figure 9. Reconstructed WZ frames](image-url)
Table 1. PSNR for WZ frame (coastguard_qcif.yuv, search-range = 64).

<table>
<thead>
<tr>
<th>SI prediction scheme</th>
<th>PSNR(dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Average</td>
<td>23.594586</td>
</tr>
<tr>
<td>Full pel-ME 4x4</td>
<td>21.586695</td>
</tr>
<tr>
<td>Full pel-ME 8x8</td>
<td>26.413514</td>
</tr>
<tr>
<td>Full pel-ME 16x16</td>
<td>28.0283</td>
</tr>
<tr>
<td>Half pel-ME 4x4</td>
<td>28.767624</td>
</tr>
<tr>
<td>Half pel-ME 8x8</td>
<td>30.579922</td>
</tr>
<tr>
<td>Half pel-ME 16x16</td>
<td>30.350176</td>
</tr>
</tbody>
</table>

The PSNR plot in the Figure 10 indicates that the half-pel ME with 8x8 block size performs better than any other schemes. The gain in PSNR using ME for prediction is approximately 7dB compared with simple frame averaging scheme.

11. Conclusions and Future Work

It can be concluded that the SI generation method proposed in this method using sub-pixel ME and MC as described in H.264 standard tools produces better results when compared to simpler methods like averaging and full-pixel ME. Also the block size of 8x8 for ME produces better results for the sequences tested.

For future work, the quarter-pixel motion estimation needs to be added to improve the SI prediction further. Intra prediction is also need to be implemented as described in section 8.

12. References