A SNR Scalability Scheme Based on the H.264/AVC Video Coding Standard

M. Mahdi Ghandi
University Of Essex
mahdi@essex.ac.uk

Mohammed Ghanbari
University of Essex
ghan@essex.ac.uk

Abstract: This paper proposes an efficient SNR scalable extension for the H.264/AVC video coding standard. The coder consists of two motion compensated sub-coders generating the base and the enhancement layers. The base layer is fully H.264 compatible while the enhancement layer bitstream has been modified. The enhancement layer coder uses the information of both layers in order to improve the coding efficiency.

Keywords: SNR scalability, H.264/AVC, layered video coding.

1 Introduction

The new H.264/AVC [1] video coding standard has been proposed recently by the Joint Video Team (JVT) of ITU-T and ISO/IEC experts. It is based on a motion compensated hybrid DCT coding similar to H.263 [2] and MPEG-4 [3] standards, but it achieves a significant improvement in rate-distortion efficiency relative to the existing standards [4-6].

Scalability is an important feature of recent video coders [2], [3] and has many applications in video streaming on wired and wireless communication channels. The first version of H.264 does not support scalability. However, scalability has been listed on the work plan as an important tool that should be supported by the standard. There are a number of proposals for scalability to be added to H.264 in the literature [7-9] and submitted standard contributions [10, 11].

Scalability is to partition a video bitstream into layers such that the base layer is independently decodable into a video sequence with reduced quality (SNR) or spatial/temporal resolution. Enhancement layers provide additional data necessary for video reproduction with higher SNR or spatial/temporal resolution.

SNR scalability, firstly proposed by Ghanbari [12], is to quantize the DCT coefficients to different levels of accuracy by using different quantization steps. Therefore, the resulting streams have different quality or SNR levels.

In hybrid video coding methods, the coefficients of the previously coded pictures can be used as a prediction of current picture by Motion Compensation (MC). In scalable scenarios, for the base layer pictures only the references of the same layer are used in MC to avoid drift, but for the enhancement layer, both the base and the enhancement data are available for prediction. In the proposed method we apply an independent Motion Estimation (ME) to predict the Macroblocks (MBs) of the enhancement layer from the previously coded enhancement layer pictures. The coder also takes advantage of the base layer data to have better compression in coding of enhancement Motion Vectors (MVs) and residual data.

The paper is organized as follows. Section 2 presents an overview of the H.264 coding method emphasizing the details that are modified for the enhancement layer. In section 3 the proposed method is discussed followed by simulation results in section 4, and finally Section 5 concludes this paper.

2 Overview of H.264

In H.264/AVC, every MB of the input picture is first predicted in intra or inter mode. In both types there are various advanced prediction modes
spatial modes for intra and temporal for inter) that are efficiently selected for each MB. The residual data between the original and the predicted blocks are then transformed, quantized and entropy coded. The DCT transform is a 4x4 integer transform and the entropy coding can be either Context Based Adaptive Variable Length Coding (CAVLC) or Context Based Adaptive Binary Arithmetic Coding (CABAC). An in-loop deblocking filter is applied for removing the blocking artefacts in the reconstructed pictures. A simplified block diagram of a typical inter-coding process is depicted in Figure 1.

Figure 1: A simplified block diagram of an H.264 encoder (for inter-coding).

2.1 Inter MB partitioning modes

In the core H264 codec, every inter-coded, 16x16 pixel MB can be partitioned into various block sizes and shapes illustrated in Figure 2. The partitioning choice of a MB into 16x16, 8x16, 16x8 or 8x8 blocks is determined by mb-type. In 8x8 mode (i.e. mb-type 3) each of the blocks can be further divided independently into 8x8, 8x4, 4x8 or 4x4 sub-partitions determined by sub-mb-type. Note that each of these blocks contains its own MV and hence more precise motion compensation can be performed when the MB is divided into smaller blocks. Moreover, every macroblock partition (but not sub-partition) shown on the top of Figure 2, could have a different reference picture which is determined by ref-idx.

3 SNR scalable H.264 codec

In the proposed method of scalable coder (illustrated in Figure 3,) the base layer is coded exactly the same as the standard non-scalable coder. In the enhancement layer, an independent ME process determines a new set of MVs. However, due to the high overhead of MVs, each MB has a prediction mode that specifies whether to send the new MV or use the base layer data.

Figure 3: Block diagram of the proposed SNR scalable coder.

3.1 Enhancement layer prediction modes

Every MB of the enhancement layer picture can be coded in intra or inter mode. When it is inter-coded, usually both base and the enhancement layers of previously coded pictures are available as well as the data of the corresponding base layer. Therefore, three different enhancement prediction modes have been designed to efficiently use the best reference for intra prediction. Each MB partition (but not sub-partition) can be predicted with one of the following modes:

- **Upward Mode**: In the upward mode no MV is sent for the block and the prediction is made by zero MVs. The reference picture is one of the base layer previously coded pictures.
- **Direct Mode**: In the direct mode, also no MV is sent but the prediction is made by the...
MV s equal to the corresponding base layer pictures’ MVs. The direct mode reference picture is one of the previously coded pictures of the enhancement layer.

- **Forward Mode:** In the forward mode, the new set of MVs is sent and the reference picture is one of the previously coded pictures in the enhancement layer.

It should be mentioned that in some enhancement layer pictures (e.g. the first frame of a sequence,) there is no reference picture. Hence, in those cases, as well as intra modes, the only possible inter prediction mode is upward mode. Additionally, in the enhancement layer, similar to the base layer, mb-type and (in 8x8 mode) sub-mb-type syntaxes determine the partitioning mode of each MB. However the sub-mb-type is sent only when the 8x8 block is in forward mode. In upward mode, since all MVs are zero, there is no need to send sub-mb-type (it is always 0). In direct mode also sub-mb-type is not sent and is equal to the corresponding value of the base layer. Note that even when a 16x16, 16x8 or 8x16 partition is in direct mode, it can have more than one pair of MVs (and also more than one reference) depending on the base layer MB division modes (and references).

After prediction with the appropriate mode, the MVs, residual data and other headers should be coded and sent. In the enhancement layer, the coding method is modified, which is described in the following sections.

### 3.2 Motion Vector Coding

In the base layer of H.264 coder, for coding an MV, it is firstly predicted from the MVs of neighbour blocks (generating PMV) and then the difference between the original MV and PMV (i.e. MVD) is calculated and coded. In the enhancement layer, when a neighbour block is in upward or direct mode or intra coded, it does not have MV. In these cases the MVs of the corresponding base layer blocks are used for prediction.

In CABAC mode, the sign and absolute value of the MVD are coded separately. For coding the sign of MVD in the base layer, since it is statistically almost equal to be negative or positive, equal probability model is used. However, in the enhancement layer MVs have a correlation to the corresponding base layer MVs and so there are 4 different context models for sign coding (two for horizontal and two for vertical MVD). The context index of MVD sign is determined by

\[
mvd\_sign\_context\_index = 2 \times hv + \text{sign}(base\_mv(hv) - PMV(hv))
\]

where \(hv\) is 0 for horizontal and 1 for vertical and base_mv is the corresponding base layer MV. It can be observed that this context depends on the base layer MV which has a high probability to have a value near the enhancement layer MV.

### 3.3 Coding of Residual Data

The residual data is calculated by subtracting the original and predicted blocks. Every 4x4 pixel block of the residual data is DCT transformed, zigzag scanned and quantized. In the base layer residual data is quantized coarsely with quantizer steps equal to QB. Hence what is actually coded in the enhancement layer is the quantization distortion of the base layer [13]. These coefficients are smaller than QB and when are re-quantized in the enhancement layer, if the quantization step (QE) is bigger than QB/2, the resulting coefficients are in the range of -1 to +1.

In H.264, when quantization parameters of the base and the enhancement layers (QPB and QPE respectively) have a difference value equal or less than 6, as shown in “Equation (2)'', the QE is equal or bigger than QB/2 [1] and then the coefficients are between -1 and +1.

\[
QPB - QPE \leq 6
\]

In practice, as a result of de-blocking filter and clipping in the reconstruction procedure, this may have some exceptions. But they can be neglected and coefficients are clipped between -1 and +1 by the coder in the described condition.

In CABAC mode for coding the residual data, the “significance map”, signs and “absolute levels” are coded and sent separately. In the condition of “Equation (2)'" is met, since all coefficients are between -1 and +1, there is no need to send the “absolute levels”. By this method a number of bits is saved in the enhancement layer without significantly degrading the quality.

### 3.4 Bitstream Structure

To implement the proposed method in the H.264 structure, the P-Picture syntax has been modified for the enhancement layer. To determine the prediction mode, we used the reference picture identifier (ref-idx). The ref-idx is equal to zero for
forward mode, 1 for upward and 2 for the direct mode. In the case of multi-reference modes, the higher reference indices point to other reference pictures.

In this modified structure, since the ref-idx determines the prediction mode and in some modes there is no need to send sub-mb-type and MVs, the ref-idx is sent before sub-mb-type to prevent sending the data that are not needed. One can modify the bitstream syntax to another structure. For example add more possible modes in mb-type and sub-mb-type instead of changing the semantic of ref-idx and it may be more sensible. However, it may have no effect on the coding efficiency which is the aim of our simulation.

4 Simulation Results

The encoder and decoder of the proposed method have been implemented using the standard JVT codec software version 7.3. It has been tested in different coding scenarios for different video test sequences. In this section first the improvement of the proposed prediction modes is verified. Secondly the results are compared with the scalability method in [8]. Finally the overall performance of the proposed method is demonstrated.

4.1 Evaluation of Prediction Modes

To design the prediction modes, we considered having two upward and forward modes as the start point. Then direct and bidirectional modes where added to see whether the improvement can be achieved. In the Bidirectional mode which is one of the prediction modes of the enhancement layer in H.263 [2], the predicted blocks of the forward and upward are averaged. In the implementation of bidirectional mode, to achieve the best possible results, we performed a separate Lagrangian optimized ME for this mode which makes the encoder almost twice more complex.

We performed the tests on different video sequences in different conditions, and some of the selected results are reported in this paper. Tables 1 and 2 show the bitrate overhead of the scalable scheme compared to the non-scalable one in various combinations of modes in different scenarios, for the “Foreman” sequence. From the Tables it can be seen that adding the bidirectional mode increases the overhead and also as mentioned before increases the complexity of the encoder, whereas adding the direct mode, reduces the overhead of the scalable coder. It is achieved by reducing the overhead caused by MVs in the enhancement layer. The same tests on different sequences like “Mobile”, “Claire” and “Mother and daughter” confirm these results.

<table>
<thead>
<tr>
<th>Available modes</th>
<th>SNR-Y(dB)</th>
<th>%overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fw,Up</td>
<td>35.63</td>
<td>39.8</td>
</tr>
<tr>
<td>Fw,Up,Bidir</td>
<td>35.63</td>
<td>39.7</td>
</tr>
<tr>
<td>Fw,Up,Drcxt</td>
<td>35.63</td>
<td>34.2</td>
</tr>
<tr>
<td>Fw,Up,Bidir,Drcxt</td>
<td>35.65</td>
<td>36.0</td>
</tr>
</tbody>
</table>

Table 2: Foreman QCIF@10Hz
QPE=28, QPB=38

<table>
<thead>
<tr>
<th>Available modes</th>
<th>SNR-Y(dB)</th>
<th>%overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fw,Up</td>
<td>35.61</td>
<td>23.6</td>
</tr>
<tr>
<td>Fw,Up,Bidir</td>
<td>35.64</td>
<td>25.2</td>
</tr>
<tr>
<td>Fw,Up,Drcxt</td>
<td>35.63</td>
<td>21.9</td>
</tr>
<tr>
<td>Fw,Up,Bidir,Drcxt</td>
<td>35.63</td>
<td>23.1</td>
</tr>
</tbody>
</table>

Table 3 shows the overhead results of the “Foreman” test sequence in various combinations of QPB and QPE. The results of the same test sequence for the proposed method in [8] are illustrated in table 4.

| QP | 25 | 30 | 35 | 40 |
|-----------------|-----------|-----------|
| 20              | 15.1      | 7.2       | 3.5 |
| 25              | 29.9      | 16.9      | 9.1 |
| 30              | .         | 37.1      | 22.9 |
| 35              | .         | .         | 43.8 |

Table 3: Bitrate overhead (in %) of proposed scalable coder compare to non-scalable Foreman QCIF@30Hz, 150frames

| QP | 25 | 30 | 35 | 40 |
|-----------------|-----------|-----------|
| 20              | 15.9      | .         | 4.2 |
| 25              | 30.0      | 16.5      | 10.4 |
| 30              | .         | 40.2      | 25.2 |
| 35              | .         | .         | 51.8 |

Table 4: Bitrate overhead of method in [8] Foreman QCIF@30Hz, 150frames

From the Tables it is clear that our proposed method outperforms the method in [8] in almost all combinations of QPB and QPE. This improvement is the result of better prediction modes design and more efficient MV and residual data coding. The improvement is more significant when the overall bit rate is low (QP is high) and then the overhead caused by the MVs are more significant, for the reason that in the proposed method the direct mode reduces this overhead.
4.2 R-D Performance

The Rate-Distortion performance of the proposed method is shown in Figures 4, 5 and 6. The results are shown for three different video sequences: “Foreman” which is a high active sequence, “Mobile” is a sequence with very detailed texture and “Claire”, a head and shoulders sequence with very low activity. Both three sequences are QCIF and coded at 10 frames/sec. The encoded bitstreams have one intra picture at the start and the rest are P-pictures. The CABAC mode is enabled for all tests and the number of reference pictures is set to one.

For every sequence, the scalable coder was tested in three different values of dQP (QPB-QPE). As was expected and is discussed in [13], when more bit rate budget is allocated to the enhancement layer (larger dQP) the efficiency of the scalable coder is better and closer to the non-scalable coder.

It should be mentioned that in these cases the quality of the base layer is poor. Another interesting point of the proposed method is that the quality degradation of the scalable coder compared with the non-scalable one is almost equal for different video sequences, whereas in some of other methods [11], it strongly depends on the input video sequence. For example, when dQP is equal to 10, the quality degradation of the scalable coder is around 1dB for different test sequences. One of the reasons of this is the existence of different possible prediction modes. They make the encoder more flexible and adaptive to different conditions. Table 5 shows the average selected modes for three different test sequences. It can be seen that the percentage of the selected modes are different for different test sequences.

Table 5: Average selected prediction modes (percent), QCIF@10Hz, QPE=28 and dQP=4 and 10

<table>
<thead>
<tr>
<th></th>
<th>%Forward</th>
<th>%Upward</th>
<th>%Direct</th>
</tr>
</thead>
<tbody>
<tr>
<td>dQP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobile</td>
<td>30</td>
<td>59</td>
<td>29</td>
</tr>
<tr>
<td>Foreman</td>
<td>35</td>
<td>68</td>
<td>21</td>
</tr>
<tr>
<td>Claire</td>
<td>40</td>
<td>72</td>
<td>16</td>
</tr>
</tbody>
</table>

5 Conclusion

We have proposed a new SNR scalable coder based on the H.264/AVC video coding standard. A new set of prediction modes is proposed for the enhancement layer as well as modifications to the coding method of motion vectors and residual data. Simulation results show that our proposed method outperforms the other comparable approaches as a result of more efficient prediction modes and also improved MV and residual data coding.
Acknowledgements

This project is supported by Engineering and Physical Sciences Research Council (EPSRC) of the UK.

References