# A Drift-Reduced Hierarchical Wavelet Coding Scheme for Scalable Video Transmissions

Roya Choupani
Computer Engineering Department
Delft University of Technology
Delft, The Netherlands
Email:roya@dutepp0.et.tudelft.nl

Stephan Wong
Computer Engineering Department
Delft University of Technology
Delft, The Netherlands
Email:J.S.S.M.Wong@tudelft.nl

Mehmet R. Tolun
Computer Engineering Department
Ç̱ankaya University
Ankara Turkey
Email:tolun@cankaya.edu.tr

***Abstract*—Scalable video coding allows for the capability of (partially) decoding a video bitstream when faced with communication deficiencies such as low bandwidth or loss of data resulting in lower video quality. As the encoding is usually based on perfectly reconstructed frames, such deficiencies result in differently decoded frames at the decoder than the ones used in the encoder and, therefore, leading to errors being accumulated in the decoder. This is commonly referred to as the drift error. Drift-free scalable video coding methods also suffer from the low performance problem as they do not combine the residue encoding scheme of the current standards such as MPEG-4 and H.264 with scalability characteristics. We propose a scalable video coding method which is based on the motion compensation and residue encoding methods found in current video standards combined with the scalability property of discrete wavelet transform. Our proposed method aims to reduce the drift error while preserving the compression efficiency. Our results show that the drift error has been greatly reduced when a hierarchical structure for frame encoding is introduced.**

***Index Terms*—Scalable Video Coding, Drift Error, Discrete Wavelet Transform**

## I. INTRODUCTION

Communications networks, both wireless and wired, offer variable bandwidth channels for video transmission[1], [2]. Display devices have a variety of characteristics ranging from low resolution screens in small mobile terminals to high resolution projectors. The data transmitted for this diverse range of devices and bandwidths have different sizes and should be stored on media with different capacity. Moreover, an encoding which makes use of a single encoded data for all types of bandwidth channels and displaying devices capacities could be of a remarkable significance in multimedia applications. Scalable video coding schemes are intended to be a solution for the Internet heterogeneity and receiver display diversity problem by encoding the data at the highest quality but enabling the transmitter or receiver to utilize it partially depending on the desired quality or available bandwidth and displaying capacities. In addition, this solution can provide the flexibility of transmitting video over heterogeneous networks such as the Internet. A scalable solution is intended to provide an encoding scheme which requires minimal processing at the server or network side and low decoding complexity at the client side. The encoding scheme needs to be comparable with the existing motion-compensated prediction and block transform encoding methods in compression efficiency. Considering that the major and most effective part of the current compression schemes in the popular encoders comes from the elimination of temporal redundancy using motion compensation, any prospective method should combine the scalability features with the motion estimation recursive loop. This requirement led researchers to consider fine granular scalable video coding methods as a solution. An implementation of these methods is the support for multilayer video coding defined in MPEG-2 standard [4]. Fine granular scalability and multilayer video coding methods however suffer from the so-called drift error problem which occurs whenever incompletely decoded frames are utilized in the reconstruction of the inter-frames. Besides, the above-mentioned methods can provide a discrete scalability property which means the receiver has to choose one of the available quality levels. Even the recent developments in scalable video coding which were published in H.264 standard by The Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG [7] are optimized forms of fine granular scalability which can provide a limited number of quality levels [8]. Many solutions have been proposed for solving the drift error and in some cases they have been able to reduce its effect to some extent at the cost of reduced coding efficiency. Drift free solutions based on discrete wavelet transforms have also been proposed which encode each frame separately as described in JPEG2000 standard[23]. The main disadvantage of these methods is the lack of integration with the recursive loop of motion compensated and block-based coding techniques which reduces their efficiency. Three dimensional wavelet transform coding provides a better efficiency but at the same time introduces the drift error. Our proposed method described in this paper falls in the group of methods based on the discrete wavelet transform. In addition, we have also integrated motion estimation and block based encoding into our scheme. The drift error has been largely reduced by introducing a hierarchical structure for the frames in a group of pictures. The hierarchical coding algorithms such as Embedded Zero Tree [5] provides the possibility of a continuous scaling the frames in bits per pixel scalability. Our proposed method provides this characteristic in block level while attempting to control the drift error. Our main contribution is the introduction of a structure which provides the feasibility of continuous scalability by combining motion estimation with scalable coding at a very low drift error effect. Our experimental results show that for a bit rate of 1000 Kbps, a peak signal to noise ratio of 38 is achievable. The remainder of this paper is organized as follows. Section II introduces the main scalable coding methods. Section III describes the details of our proposed method. In Section IV, we provide the experimental results and finally, in Section V, we draw the conclusions.

## II. RELATED WORK

Due to the heterogeneity in current-day networks (especially on the Internet), the unpredictability of traffic loads, and the varying delays on the client side, it is impossible to correctly determine a specific bit rate for a video stream. Consequently, the encoder should either consider the lowest possible bit rate that guarantees delivery without delay or choose an encoding scheme which can adapt with the fluctuations in the bit rate range. This means that it should be possible to partially decode the video stream at the incoming bit rate at the video quality associated with that bit rate. A solution to this problem is encoding the video data in a rate scalable form enabling adaptation to the receiver or network capacities. Increasing the video quality gradually is the common characteristic of all scalable video coding schemes. The quality increase is accomplished though the gradual increased availability of the data ”units” that were encoded in a granular manner. These video coding schemes are commonly referred to using the term Fine Granularity Scalability (FGS) [11], [12]. It is clear that a gradual increase in the frame size, bit rate, or frame rate is achieved through adapting the granularity of a stream to the bit rate capability of the network. A fine granularity scalability scheme defines the video content in a multi-layer format [13], [14]. A higher quality for a video is attained by increasing the number of layers decoded at the receiver side. Bit-rate or signal-to-noise ratio (SNR) scalability is a technique to code a video sequence into two or more layers at the same frame rate and the same spatial resolution, but utilizing different quantization accuracy. Figure 1 depicts the SNR scalability decoder defined in MPEG-2 video-coding standard [15], [18]. The base layer bitstream is decoded by the base layer variable-length decoder (VLD) first. The inverse quantizer in the base layer produces the reconstructed DCT coefficients. The enhancement bitstream is decoded by the VLD in the enhancement layer and the enhancement residues of the DCT coefficients are produced by the inverse quantizer in the enhancement layer. A higher accuracy DCT coefficient is obtained by adding up the base-layer reconstructed DCT coefficient and the enhancement-layer DCT residue. An important issue here is if the encoder uses the enhancement-layer information in the motion-prediction loop and the enhancement-layer information is not received by the decoder, drift error is introduced in the base layer. The drift error problem which is common to all multi-layer scalability schemes has been attacked by several researchers. Mayer, et. al., [19] introduces drift compensation value D which is added to the prediction error prior to quantization. When the D value equals the accumulated error, the drift error is eliminated. When the D value equals zero, the drift error is full present. Consequently, a good balance between the quality of the reconstructed image when the enhancement layer is utilized and when only the base layer (with D value) is utilized should be determined. In drift clipping, the drift is dynamically controlled by comparing to a maximum value D [19]. A good choice for D can be determined by utilizing the base-layer quantizer step size. Consequently, the decoder should utilize the same clipping parameters resulting in that these must be occasionally sent as additional information. Drift leaking [22] limits the accumulation of drift by multiplying the accumulated value D by a leak coefficient. The selection of an appropriate drift coefficient is application dependent. H.264 tries to control the drift error by introducing a new concept called key pictures [3]. Key frames are not necessarily intra-coded frames. For each key picture a flag is transmitted, that signals whether only the base layer information or together with the enhancement layer information was utilized in the reconstruction of reference frames in the motion estimation. By introducing a hierarchical reference frame organization, H.264 allows all enhancement layer frames to utilize the references with the highest available quality for motion estimation, which enables a high coding efficiency for these key pictures [21]. Drift error is not eliminated completely but its effect is minimized and limited to the frames between two consecutive key pictures [16]. The tradeoff between enhancement layer coding efficiency and drift error can be adjusted by the choice of the number of frames between two consecutive key pictures or the number of hierarchy stages. Figure 2 depicts a sample enhancement layer frame reference organization in H.264 [17]. The discrete wavelet transform (DWT) has been used for scalable encoding of still image and video multimedia as described in JPEG2000 [23]. Two low and high pass filters are used for separating low and high frequency data, respectively. Spatial oriented trees such as Embedded Zero Tree Wavelets (EZW) [5] and Spatial Partitioning in Hierarchical Trees (SPIHT) [9] are used for organizing wavelet coefficients in order of importance for scalability. A client may perform a partial reconstruction by truncating the less

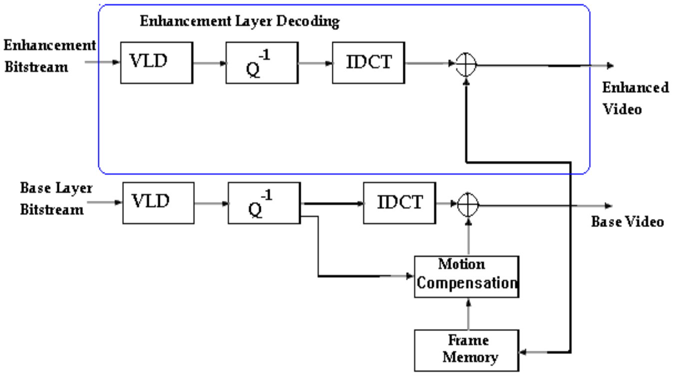

Fig. 1. SNR scalability decoder block diagram

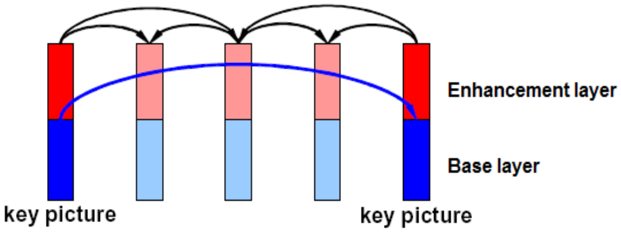

Fig. 2. Key picture concept of H.264 for hierarchical prediction

important coefficients from the end of the stream. These methods, however, suffer from a low performance problem since they do not include a motion compensation feedback loop. Three dimensional wavelet transforms are used for exploiting temporal redundancy in video multimedia [6]. If motion compensated temporal filtering is used in combination with a two dimensional spatial wavelet, it is called a three dimensional (3D) or 2D+t wavelet transform. Adjusting frames along the motion vectors in a motion trajectory, may cause some regions to be left uncovered. These regions cannot be reconstructed during the synthesis phase of wavelet transform. Motion compensated implementation of 3D+t wavelet transform still suffers from the drift error problem as well[10].
Our proposed method differs from the existing methods by combining the block-based motion compensation scheme with discrete wavelet transform scalability and hierarchical reference frame selection. Block based motion estimation has the advantage of temporal redundancy reduction but when used with scalable video coding, drift error is caused. Hierarchical reference frame selection reduces the error accumulation by reducing the number of frames in a reference chain. The discrete wavelet transform lets us to have a continuous scalability.

## III. Our Proposed Method

Our proposed method involves using the scalability features of discrete wavelet transforms. To achieve a high performance which is comparable to non-scalable codes, block-based motion estimation is used. The drift error is reduced considerably by applying a hierarchical structure. The general view of the proposed structure and method are depicted in Figures 3 and 4, respectively. The frames in a group of pictures (GOP) are organized in pairs and the second frame in each pair is divided into macroblocks, motion compensated, and compressed using a discrete wavelet transform. These frames are indicated with a yellow color in Figure 3. The wavelet coefficients of each macroblock are quantized and zig-zag scanned before transmission to put them in information importance order. This makes it possible for a receiver or a network node to truncate parts of data in case of low bandwidth or processing power. Embedded zero trees (EZW) are used to arrange the coefficients. The first frames of the pairs from the first level are grouped in the next level of the hierarchy in pairs and the same division into blocks, motion compensation, and wavelet encoding steps are applied to them. This means the first frames of the pairs which serve as the reference frames for the second frames at the same pair, are processed at a higher level where they are finally positioned as the second frame of a pair. This procedure is repeated at the following levels of the tree hierarchy. The wavelet transform using Daubechies D4 scaling and wavelet functions as given in Equations 1 and 2 are applied to the residue of the macroblocks after motion estimation.

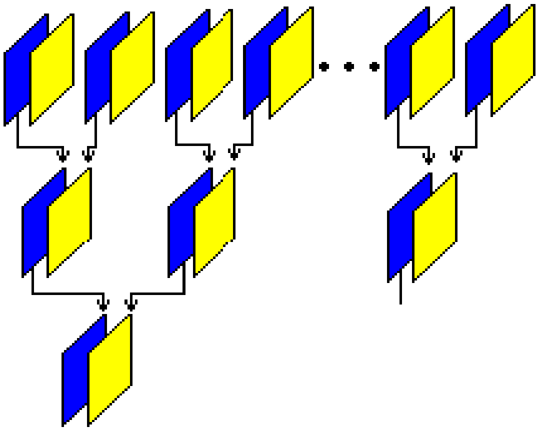

Fig. 3. The proposed hierarchical structure

$$a_i = s_0 s_{2i} + h_1 s_{2i+1} + h_2 s_{2i+2} + h_3 s_{2i+3} \tag{1}$$

$$c_i = g_0 s_{2i} + g_1 s_{2i+1} + g_2 s_{2i+2} + g_3 s_{2i+3} \tag{2}$$

where $s_{ij}$ indicates the data values, $h_{1-4}$ are scaling coefficients and $g_{1-4}$ are the wavelet function coefficients. The steps followed in the proposed method are depicted in Figure 4. Since pairing the frames is required before motion estimation and applying wavelet transform, a buffer is introduced to store the frames of a GOP. An optimization is proposed for reducing the size of this buffer which is discussed in Section IV.

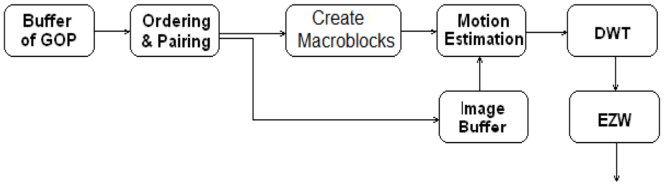

Fig. 4. Block diagram of our proposed method

### A. Drift error reduction

If only the lowest layer of the hierarchy is considered, the drift error is limited to one frame as the second frame at each pair is motion compensated and encoded using the first frame of the same pair. However, any accuracy change in the second layer affects the first frames of each pair in the lowest layer and therefore the error is accumulated. The worst case situation is when error is introduced in the topmost layer of the hierarchy which affects the whole tree. However, in this case the number of frames in a series of frames referencing each other is limited to the tree height and, therefore, the GOP size and hence tree height should be determined in a tradeoff with the max tolerable drift error. This structure reduces the drift error in a logarithmic manner. The proposed structure falls in the group of non-delay methods where no frame need to be buffered till the arrival of the following frame(s) for decoding. This makes the decoder implementation simple, with minimum memory requirement.

### *B. Bit rate scalability*

Bit rate scalability in the available scalable video coding standards such as MPEG-4 or H.264 is defined as multi-layer coding of the frames where the number of layers determines the granularity of the video. The multi-layer bit rate scalability suffers from the problem that using enhancement layer data in motion compensation loop may result in drift error. On the other hand motion compensation using base layer only can reduce compression efficiency. For example, the fine granularity quality scalable (FGS) coding in MPEG-4 was chosen in a way that drift is completely omitted by using base layer frames as reference frames in motion compensation and, therefore, any loss or modification of a quality refinement packet does not have any impact on the motion compensation loop. The number of layers in these standards should also be limited as carefully designed since the multi-layer concept for quality scalable coding becomes less efficient, when the relative rate difference between successive layers gets smaller. As our proposed method is based on discrete wavelet transform, multi-layer restrictions are not effective here. The number of bits used for representing wavelet transform coefficients of the motion compensated residues is reduced for a lower bit rate transmission over a low bandwidth channel. This goal can also be achieved by organizing the wavelet coefficients in a spatial tree structure such as EZW[5] or SPIHT[9] in which cases, the transmitted bitstream is truncated considering the bandwidth capacity of the channel.

### *C. Spatial scalability*

Scalability in frame resolution was introduced with the MPEG-2 standard [4]. A multi-layer structure was considered for implementing the scalability where the base layer contained the lowest resolution and a higher resolution frame was reconstructed by upsampling the base layer data and adding the refinements from upper layers. The main problem with this method is integrating it with the motion compensation loop. Using high resolution frames for motion estimation can reduce the compression rate when only a low resolution frame is reconstructed at the receiver side. Motion estimation in low resolution frames on the other hand causes drift error problem as the difference frames obtained from low resolution base frames do not include any information about the eliminated rows and columns. One way to reduce the drift error is by limiting the length of a group of pictures in a series of motion compensated frames without reducing the compression efficiency. Our proposed method performs this by organizing the frames in a hierarchy and therefore is suitable for these types of scalability applications.

### *D. Temporal scalability*

Temporal scalability in the traditional video coding methods is achieved through placing some of the frames in the base layer and the rest in the enhancement layer(s). An example is dividing the frames of a GOP by putting even numbered frames in base layer and odd numbered ones in the enhancement layer. A drift error problem will appear if the motion compensation involves the frames of the enhancement layer, if the receiver is capable of reconstructing the video in a lower frame rate. In our method a 50% temporal scalability is achieved by dropping the second frames of each frame pair at the lowest level of the hierarchy. This scalability is accomplished without any drop in the compression efficiency or drift error problem. A higher rate of scalability is possible by eliminating the second frames at the next level. The number of levels in the tree hierarchy is not an upper or a lower limit on the temporal scalability. This fact is described by considering the characteristic of the proposed hierarchy where the frames of a series are not chained together in a linear structure. This means that any possible rate of temporal scalability is achievable by eliminating only some of the second frames of the frame pairs.

The temporal scalability achievable at each level can be expressed using Equation 3.

$$SR_n = \sum_{i=1}^{n} \frac{1}{2^i} \tag{3}$$

where $SR_n$ refers to scalability rate at level n.

## IV. EXPERIMENTAL RESULTS

Three series of video frames have been used for evaluating the proposed method. The following subsections describe our main considerations in implementing the proposed method.

### *A. Organizing Frames*

As we have to put the frames in pairs prior to applying the blocking and wavelet transform steps and repeat the procedure at the upper layers, a large memory buffer is required to temporarily store the frames. This condition is not acceptable and applicable in many hand-held devices with small memory capacity. To solve the problem we considered the following procedure for processing the frames. Figure 5 depicts a sample tree created for a GOP. For simplicity we assumed 16 frames in the GOP. The algorithm uses a stack to temporarily store the frames for motion estimation. Since always in a pair of frames, the second frame is divided into blocks and motion compensated, we check the sequence number of the arriving frame. If it is an even number, the frame is pushed into the stack. However, if the sequence number is odd, a frame is popped from the stack and used as the reference frame for motion compensating the current frame. To control the correct pair matching of the frames for motion compensation, a level

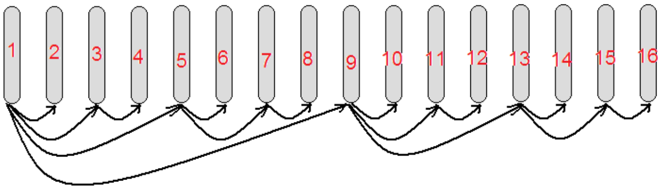

Fig. 5. The hierarchical structure of a GOP frames

value is considered which is pushed to the stack together with the frame. The level number corresponds to the tree level in our proposed hierarchical structure. The procedure is repeated at the upper layers using the frames pushed into the stack however, the even and odd sequence numbers are obtained considering the current level of the tree as described in Algorithm 1. The difference frames generated after applying the wavelet transform to residues of motion compensated are put in a transmission stream and sent whenever they are at the beginning of the stream. This step is also described in Algorithm 1.

The receiver buffer aditionally needs to store some frames during reconstruction. We store one frame for each tree level in the buffer. This is justified as the branches are reconstructed from left to right and therefore never more than one frame per level is required during reconstruction. This observation restrict the buffer size to $\log_2 GOP$.

A second important consideration regards the artifacts observed on the boundaries of the macroblocks after reconstruction. This effect especially is more serious when the wavelet coefficients are truncated or partially eliminated. We utilized post-filtering to reduce this effect as described in [24].

### B. Test Setup

To compare the efficiency of the proposed method we have used three video sequences which have been given in Table I.

TABLE I
VIDEO SEQUENCES USED FOR PERFORMANCE EVALUATION.

| Sequence Name | Resolution | Frame rate | Size |
|---|---|---|---|
| Foreman | $352 \times 288$ | 30 | 300 |
| Stefan & Martin | $768 \times 576$ | 30 | 300 |
| City | $704 \times 576$ | 60 | 600 |

**Algorithm 1** Frame Encoding Algorithm

```
StreamIndex $\leftarrow$ 1
FrameIndex $\leftarrow$ 1
NFrame $\leftarrow$ GetNextFrame()
InsertToFrameStreamAt( FrameIndex )
repeat
  NFrame $\leftarrow$ GetNextFrame()
  FrameIndex $\leftarrow$ SequenceNumber( NFrame )
  if FrameIndex is odd then
    Push(NFrame,1)
  else
    $\langle$ RefFrame,level $\rangle$ $\leftarrow$ Pop()
    MotionCompensate( RefFrame , NFrame )
    ApplyWavelet()
    InsertToFrameStreamAt( FrameIndex )
    Push(RefFrame,level)
  end if
  repeat
    $\langle$ Frame1,level1 $\rangle$ $\leftarrow$ Pop()
    $\langle$ Frame2,level2 $\rangle$ $\leftarrow$ Pop()
    if NOT StackError AND level1 = level2 then
      FrameIndex $\leftarrow$ SequenceNumber( Frame2 )
      MotionCompensate( Frame1 , Frame2 )
      ApplyWavelet()
      InsertToFrameStreamAt( FrameIndex )
      Push(Frame1,level1+1)
    else
      Push(Frame1,level1)
    end if
  until level1 $\langle\rangle$ level2 OR StackError
  while Exist( StreamFrameAt( StreamIndex ) ) do
    SendFrame( StreamFrameAt( StreamIndex ) )
    Increment( StreamIndex )
  end while
until End of GOP reached
```

To further emphasize on the differences in compression efficiencies between traditional motion compensated methods and our proposed method, we considered using CABAC entropy coding introduced with H.264/AVC standard [20]. The number of active reference frame is chosen as one throughout the experiments which means all blocks of a given frame are using the same reference frame. Peak Signal to Noise Ratio (PSNR) between decoded frames pixels and the original frame has been used as the metric for evaluating the performance of our method. We have defined PSNR as given in Equation 4.

$$PSNR = 20\log_{10}\frac{Max_I}{MSE} \tag{4}$$

$$MSE = \frac{1}{3mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} \parallel I(i,j) - I'(i,j) \parallel \tag{5}$$

where $Max_I$ indicates the largest possible pixel value, I is the original frame and I' is the decoded frame at the receiver side. Figures 6 and 7 depict the comparative results for the video sequences given in Table I.

Figure 6 depicts the effect of increasing the length of GOP on the performance of the method. As shown in the graph, a PSNR gain of one or higher dB can be achieved with relatively low bit rates. This is an indication of the fact that our proposed hierarchical structure for the frame organization does not show any major loss of efficiency gain with increasing the distance between a frame and its reference. Our experiments show that

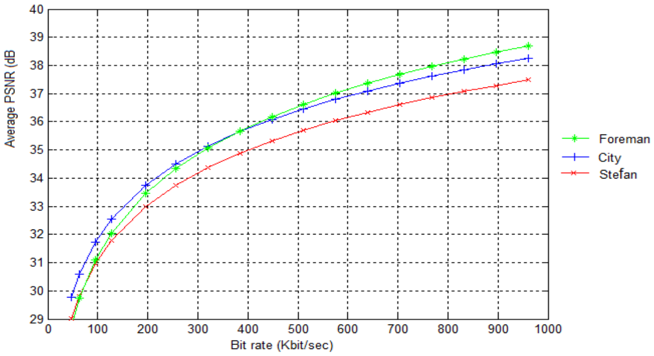

Fig. 6. PSNR at different bit rates

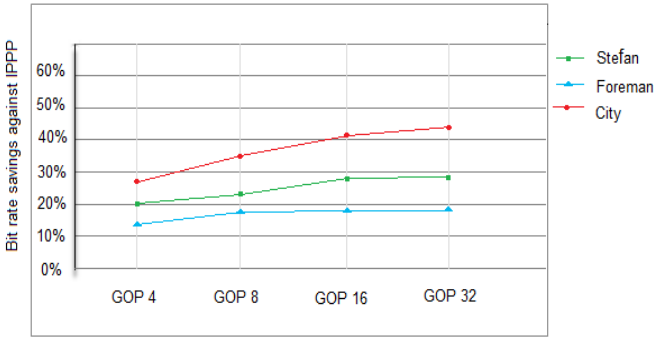

Fig. 7. Bit rate saving against GOP sizes

assuming a GOP between 16 and 32 gives the best result from the PSNR point of view.

## V. CONCLUSION

A new scalable video coding method capable of combining the motion compensated feedback loop with a three dimensional wavelet transform has been introduced. The method which falls in the group of methods with drift error effect can be used for bit rate, spatial and temporal video scaling. The method tries to reduce the drift error effect by re-organizing the frames in a hierarchical structure so that the accumulated error is limited to $\log_2(n)$, where n is the number of frames in a group of pictures. This at the same time provides the possibility of having long GOP groups for better efficiency. The experimental results shows that the compression rate of the proposed method is only slightly below the traditional motion compensated standards and the accumulated drift error is negligible. Still better results are achievable if optimizations like multiple reference frame are permitted however, we have not considered optimizing the proposed method for low memory capacity and narrow-bandwidth applications.

## REFERENCES

[1] G. Conklin, G. Greenbaum, K. Lillevold, A. Lippman, and Y. Reznik, ”Video Coding for Streaming Media Delivery on the Internet”, IEEE Trans. Circuits and Systems for Video Technology, March 2001.

[2] D. Wu, Y. Hou, W. Zhu, Y.Q. Zhang, and J. Peha, ”Streaming Video over the Internet: Approaches and Directions”, IEEE Transactions on Circuits and Systems for Video Technology, March 2001.

[3] H. Schwarz, D. Marpe, and T. Wiegand, ”Overview of Scalable Video Coding Extension of the H.264/AVC Standard”, IEEE Transaction on Circuits and Systems for Video Technology, Vol. 17, No. 9, September 2007.

[4] B. G. Haskell, A. Puri, and A. N. Netravali, ”Digital Video: An Introduction to MPEG-2”, New York: Chapman and Hall, Sept. 1996.

[5] J. M. Shapiro, ”Embedded image coding using zerotrees of wavelets coefficients”, IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3445-3462, December 1993.

[6] B. Felts, B. Pesquet-Popescu, ”Efficient context modeling in scalable 3D wavelet-based videocompression”, Proceedings of IEEE International Conference on Image Processing pp. 1004–1007 vol. 1, 2000.

[7] D. Marpe, T. Wiegand, and G. J. Sullivan, The H.264/MPEG-4 advanced video coding standard and its applications, IEEE Commun. Mag., vol. 44, no. 8, pp. 134144, Aug. 2006.

[8] G. J. Sullivan, H. Yu, S. Sekiguchi, H. Sun, T. Wedi, S. Wittmann, Y.-L. Lee, A. Segall, and T. Suzuki, New standardized extensions of MPEG-4-AVC/H.264 for professional-quality video applications, presented at the ICIP, San Antonio, TX, Sep. 2007

[9] A. Said and W. A. Pearlman, ”A new, fast, and efficient image codec based on set partitioning in hierarchical trees”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 243–250, June 1996.

[10] J. R. Ohm, ”Three-dimensional subband coding with motion compensation”, IEEE Trans. Image Processing, vol. 3, pp. 559571, Sept. 1994.

[11] M. Ghanbari, ”Two-layer coding of video signals for VBR networks”, IEEE Journal of Selected Areas Communications, vol. 7, pp. 771781, June 1989.

[12] H. Gharavi and M. H. Partovi, ”Multilevel video coding and distribution architectures for emerging broadband digital networks”, IEEE Transaction on Circuits and Systems for Video Technology, vol. 6, pp. 459-469, October 1996.

[13] H. Jiang, ”Experiment on post-clip FGS enhancement”, ISO/IEC JTC1/SC29/WG11, MPEG00/M5826, March 2000.

[14] W. Li and Y. Chen, ”Experiment result on fine granularity scalability”, ISO/IEC JTC1/SC29/WG11, MPEG99/M4473, March 1999.

[15] D. Wilson and M. Ghanbari, ”Exploiting interlayer correlation of SNR scalable video”, IEEE Transaction on Circuits and Systems for Video Technology, vol. 9, pp. 783-797, August 1999.

[16] A. Segall, CE 8: SVC-to-AVC Bit-Stream Rewriting for Coarse Grain Scalability, Joint Video Team, Doc. JVT-V035, Jan. 2007.

[17] M. Wien, H. Schwarz, and T. Oelbaum, Performance analysis of SVC, IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 9, pp. 11941203, Sep. 2007

[18] R. Mathew and J. F. Arnold, ”Layered coding using bitstream decomposition with drift correction”, IEEE Transaction on Circuits and Systems for Video Technology, vol. 7, pp. 882-891, December 1997.

[19] C. Mayer, H. Crysandt, and J.R. Ohm, ”Bit plane quantization for scalable video coding”, Proc. SPIE, Visual Commun. Image Process., vol. 4671, pp. 1142-1152, 2002.

[20] D. Marpe, G Blttermann and T Wiegand, Adaptive Codes for H.26L, ITU-T SG16/6 document VCEG-L13, Eibsee, Germany, January 2001

[21] H. Kirchhoffer, H. Schwarz, and T. Wiegand, CE1: Simplified FGS, Joint Video Team, Doc. JVT-W090, Apr. 2007.

[22] P. Amon, K. Illgner, and J. Pandel, ”SNR scalable layered video coding”, presented at the Int. Packet Video Workshop, Pittsburgh, PA, 2002.

[23] ISO/IEC JTC1/SC29/WG1 N1890R, (ISO/IEC FDIS15444-1),”JPEG2000 Part 1 Final Draft International Standard”, September 25, 2000.

[24] N. Vasconcelos F. Dufaux, ”Pre and post-filtering for low bit-rate video coding”, Proceedings of the 1997 International Conference on Image Processing (ICIP ’97) Vol. 1, pp. 291–294