FINE-GRAINED VECTOR-LESS SCALABLE VIDEO CODING FOR SPACE APPLICATIONS

Trevor Spiteri, José Luis Nuñez-Yañez

Department of Electrical and Electronic Engineering
University of Bristol, Bristol, BS8 1UB, United Kingdom
Email: trevor.spiteri@bristol.ac.uk, j.l.nunez-yanez@bristol.ac.uk

ABSTRACT

This paper presents a fine-grained scalable video coding method that is suitable for space application. The method does not transmit motion vectors explicitly, and uses wavelets and bit plane coding for video coding in a manner similar to that used by Embedded Block Coding with Optimal Truncation (EBCOT) and ICER for image coding. Video sequences take large amounts of bandwidth to be transmitted losslessly. Using scalable video coding, all the information can be compressed and stored losslessly in storage. A lossy subset can then be transmitted, and refinements can then be transmitted on request to improve the quality up to lossless. Motion estimation is performed in the base layer that is the minimum quality that can be extracted from the sequence.

1 INTRODUCTION

Lossy compression is becoming important in space applications because of the increasing amount of data and the high bandwidth costs. Yu et al. [1] present an overview of image compression in space missions, where images are most of the time stored and then transmitted to ground later on. The paper also suggests that methods using the Discrete Cosine Transform (DCT) [2] are becoming less popular because of the increasing popularity of methods based on wavelet transforms [3]. Wavelet transform methods started to gain popularity for image compression after the development of the embedded zerotree wavelet algorithm (EZW) [4]. Later developments led to methods based on set partitioning in hierarchical trees (SPIHT) [5] and on independent embedded block coding with optimized truncation (EBCOT) [6]. The EBCOT algorithm was adopted and further adapted for JPEG 2000 [7]. The ICER progressive wavelet image compressor [8], used by the Mars Exploration Rover in 2004, uses techniques very similar to EBCOT and JPEG 2000.

An important property of these techniques is their progressive nature, which means that as the compressed bitstream is progressively transmitted, the receiver can reconstruct images of successively higher quality, and the quality may go up to lossless if the bitstream is fully transmitted. This works very well with the store-and-forward mechanism, whereby the compressed data is stored for later transmission to ground. A subset of the stored data can then be transmitted. When this subset is inspected, a user might decide that a better version is required, and will transmit back a request for more data to refine the quality as much as required, up to lossless quality.

In this paper, a method for fine-grained scalable video coding that uses techniques similar to those used in image coding is presented. Scalable video coding is the coding of video at different quality levels, different resolution, or different frame rates. For the scalable video coding method to be fine-grained, there needs to be a mechanism for very gradual degradation of the quality. Taubman [9] presents some of the issues present in successive refinement of video. This paper presents a fine-grained scalable video coding method that makes it trivial to transmit a subset of the full-quality bitstream, using techniques similar to those used in progressive image coding methods like EBCOT and ICER. This makes it possible for a compressed video sequence to be partially transmitted to ground, and if required, refinements are later transmitted without the need to transmit what has already been transmitted.

Secker and Taubman [10] present a scheme by which they scale both the video data and the motion information for good performance over a wide range of bit rates. In the proposed work, motion information is derived from a basic quality layer, which is not itself scalable like the motion information in [10].

Existing work includes the Scalable Video Coding (SVC) extension of the H.264/AVC standard [11]. It is worth noting that the SVC standard is quite complex, and that the Joint Scalable Video Model (JSVM) [12] reference implementation of SVC supports coarse-grained and medium-grained scalability but no fine-grained scalability.

Section 2 describes the background for the presented work, and Section 3 presents the video coding scheme itself. Section 4 presents experimental results demonstrating the performance of the scheme, and compares the performance to the H.264/AVC JM reference implementation [13] to provide a general idea of the compression performance of the scheme. Finally, Section 5 draws the conclusions.

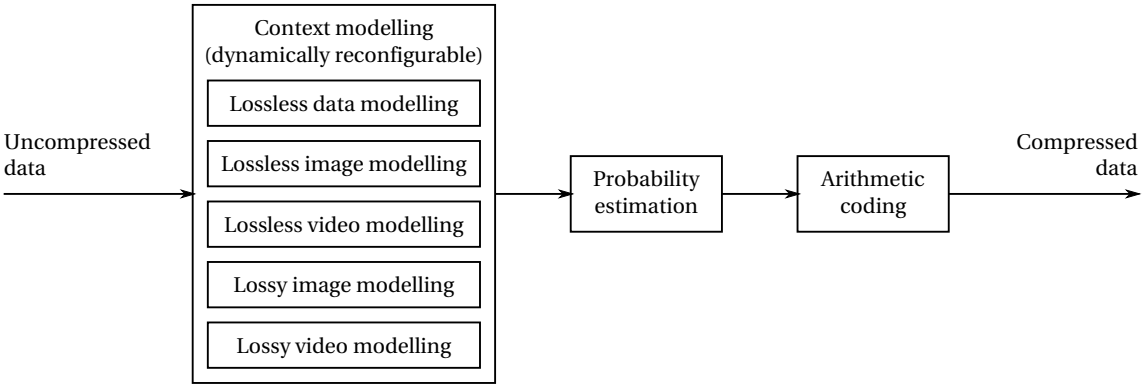

Fig. 1: Overview of universal compression system.

2 BACKGROUND

The scalable video coding scheme presented here is part of a larger project which combines video compression with image compression and generic data compression on a single dynamically-reconfigurable FPGA platform.

2.1 Universal Compression System

To enable compression of different kinds of data without having full implementations for each algorithm, the compression process can be split into multiple stages. Nuñez-Yañez et al. [14] suggest splitting the process into three stages:

1. context data modelling,
2. probability estimation, and
3. arithmetic coding.

Fig. 1 shows an overview of the suggested universal compression system. The first stage, context modelling, can be dynamically reconfigured according to the kind of data being compressed, while the other stages are the same for all kinds of data. A description of context data modelling for generic data, as well as a description of the probability estimation and arithmetic coding stages, can be found in [15]. A description of the first stage for lossless image compression can be found in [16], and a description of the first stage for lossless video compression can be found in [17]. This works focuses on progressively lossy video compression.

2.2 Wavelet Transform

The presented video coding scheme draws influence from wavelet-based image compression systems, namely EBCOT [6] and ICER [8]. The image data is transformed using two-dimensional (2-D) wavelet decomposition. Fig. 2 shows three levels of 2-D decomposition, which results in ten sub-bands. The wavelet transform is obtained using one of two lifting filters: either the reversible Le Gall 5-tap/3-tap filter [18] or the Daubechies 9-tap/7-tap filter [19]. The 9/7 filter may provide better compression at a cost: it requires fixed point multiplication, which is more difficult to implement in hardware than the shift and add operations required by the 5/3 filter, and also, using the 9/7 filter means that the image can not be decoded losslessly, as the 9/7 filter is not reversible.

2.3 Bit-Plane Coding

After the image data is transformed using the wavelet transform, the transformed data is bit-plane coded. The data in each sub-band in the transformed image is split into a number of bit planes: a plane containing the most significant bit of each value, a plane for the second most significant bits, and so on down to the plane containing the least significant bits. Whereas in EBCOT the planes are further subdivided into fractional bit planes, in ICER this is avoided to simplify the implementation, and this paper follows ICER's simplification. The bit planes from different sub-bands are then sorted such that a bit plane that should give a better quality improvement in relation to its length is preferred to another bit plane which gives less quality for the length. If only a subset of the bit planes is to be transmitted, the planes with the least quality improvement per length are dropped.

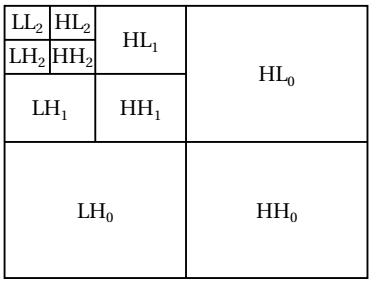

Fig. 2: The ten sub-bands from three levels of 2-D wavelet decomposition.

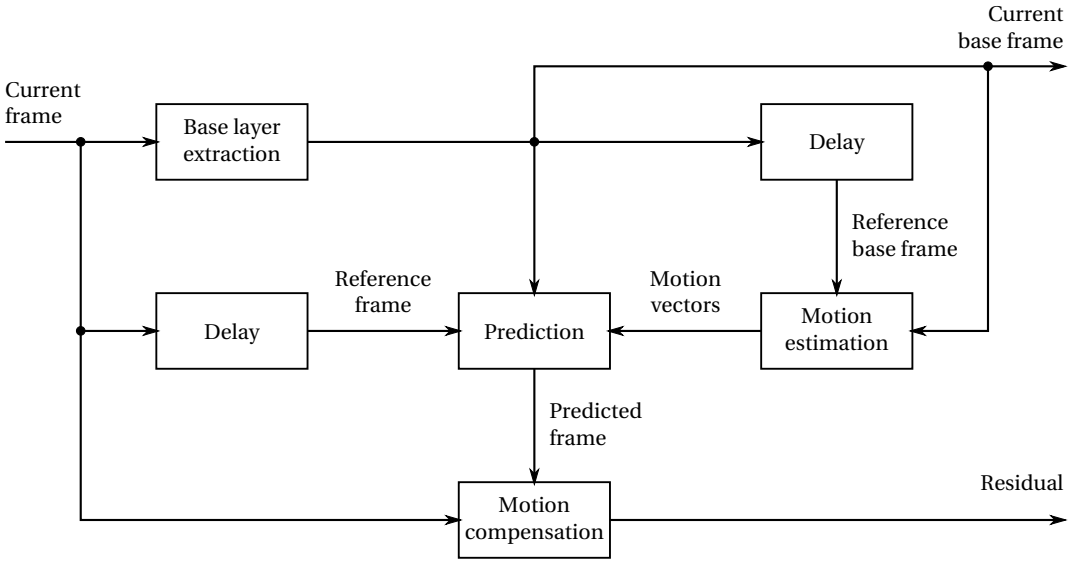

Fig. 3: Overview of the video encoder before bit-plane encoding.

Context modelling is then used to encode the bit planes. For each bit, a context is chosen according to the neighbouring pixels and more significant bits for the same pixel. After the context is selected, an adaptive arithmetic coder is used to encode the bits from the bit planes.

3 SCALABLE VIDEO CODING ALGORITHM

The scalable video coding scheme presented in this work uses temporal motion compensation to take advantage of temporal redundancy. Each frame is first transformed to the wavelet domain. Then motion compensation is performed as described in Section 3.1. Then, the output from the motion compensation stage is bit-plane encoded, and the bit planes further processed using context modelling, probability estimation and arithmetic coding, the three stages of Section 2.1.

3.1 Motion Compensation

Fig. 3 shows an overview of the video encoder before bit-plane encoding, and Fig. 4 shows an overview of the corresponding decoder following bit-plane decoding. In the first frame of a group of pictures, since there are no reference frames with which to compare, no motion estimation and compensation is performed. Instead, the frame is simply bit-plane encoded as is. The bit planes are ordered according to their quality contribution in relation to their length, and a number of bit planes is selected to be encoded as the basic quality layer. The selection of which bit planes to use for this base layer is only performed once in the first frame of the group of pictures, and in subsequent frames, the same bit planes are used to form the base layer. The base layer is not motion compensated in any frame; it is encoded just like an image would be, with the difference being that only a specified subset of the bit planes are encoded.

For the next frames, a motion estimation search is performed on the image data that can be decoded from the base layer, and a set of motion vectors are obtained. Note that since the base layer is transmitted as is, the decoder will have

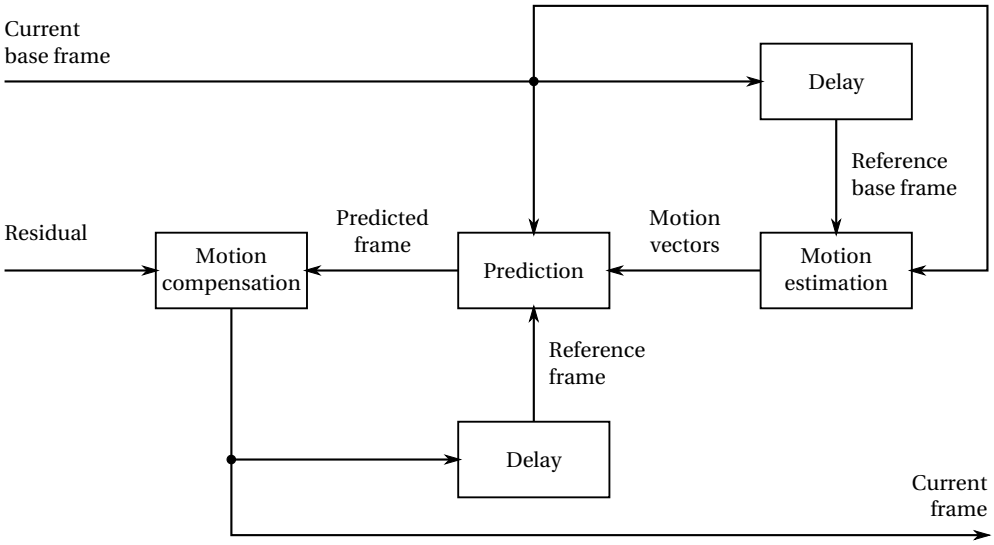

Fig. 4: Overview of the video decoder following bit-plane decoding.

access to the same base layer information as the encoder, so that the motion vectors do not need to be transmitted; the decoder can perform the motion estimation search itself to regenerate the motion vectors.

The motion vectors are then used to generate a predicted frame from a reference frame. The prediction is then subtracted from the actual frame values, resulting in the residual. Note that this is performed in the wavelet domain, not in the space domain, to enable scalable video by dropping bit planes just like bit planes are dropped in EBCOT and ICER for progressive image coding.

As can be seen in Figs. 3 and 4, the motion compensation block has access to the base layer image data. Since the base layer contains the most significant bit planes of a number of sub-bands, this information can be used to generate a better prediction of the current frame.

Sometimes the residual can be more expensive to encode than the uncompensated values. In order to mitigate this, each sub-band is split into blocks of 16×16 values each, and a decision is made per block whether to use the residual or the actual values. The decision is made by comparing the sum of the squares of the values, that is, by comparing the power. If the residual power is greater than the power of the actual values, the actual values are used instead of the residual. A single bit per block is then inserted in the final bitstream to indicate whether to use the residual or the actual values.

3.1.1 Motion Estimation Search

The motion estimation search on the base layer values can be either performed in space domain or in wavelet domain. For the search to be performed in space domain, the base layer bit planes, which are in wavelet domain, have to be transformed back to space domain before the search. Then, a search similar to that in [17] is used.

When the search is done in wavelet domain, the base layer does not need to be transformed to space domain. The motion estimation search in wavelet domain is different to that performed in space domain. An anisotropic double-cross search based on that presented in [20] is used. Two different motion estimation methods were investigated in the wavelet domain. The first is pixel-based, that is, a search is performed once for every pixel. A search window around the pixel is used to enable the pixel-based search, in a similar way to [17]. The second method is macroblock-based, whereas a search is performed using a block of values across different sub-bands as proposed by [20]. These two methods, as well as the space-domain search, will be compared further on in Section 4.

In each of the three methods mentioned, no sub-pixel motion estimation was used. This means that the motion vectors resulting from the search are a whole number of pixels of motion in the original space-domain frame. However, in the wavelet domain, interpolation is still required, because one pixel of motion in space domain translates to one half a pixel of motion in sub-bands HL_0, LH_0 and HH_0 (see Fig. 2), one quarter of a pixel of motion in sub-bands HL_1, LH_1 and HH_1, and so on. Because of this, when generating a prediction, interpolation is used to obtain the predicted values from the reference frame which is in wavelet domain.

3.2 Bit-Plane Ordering

Following the motion compensation stage, the residual for each frame consists of a number of sub-bands, each with a number of bit planes. Each of these bit planes is further processed using the context modelling, probability estimation, and arithmetic coding stages mentioned in Section 2.1. The length of the output bitstream for each plane is recorded, together with the number of non-zero bits in the plane, which is useful in determining the distortion that would result from dropping the plane.

A distortion metric similar to that used in EBCOT [6] is used for ordering the bit planes. The estimated distortion D_n for dropping a bit plane n in a sub-band B is given by

$$D_n = w_B^2 \sum_{\mathbf{k} \in B} (\hat{s}_n[\mathbf{k}] - s[\mathbf{k}])^2, \tag{1}$$

where w_B is the L2-norm of the wavelet basis function for sub-band B, $\hat{s}_n[\mathbf{k}]$ is a value in the sub-band after dropping the bit plane, and $s[\mathbf{k}]$ is a value in the sub-band if the bit plane is not dropped. For the least significant bit plane, $n = 0$, and (1) can be written as

$$D_0 = w_B^2 \sum_{\mathbf{k} \in B} (\hat{s}_0[\mathbf{k}] - s[\mathbf{k}])^2. \tag{2}$$

For this plane, a non-zero bit at position $\mathbf{k}$ will result in an estimation error of magnitude 1, so the summation term will be equal to the number of non-zero bits in the plane. If there are h_0 non-zero bits, we can say that $D_0 = w_B^2 h_0$. For the plane $n = 1$ containing the second least significant bits, each non-zero bit will result in an estimation error of magnitude 2, contributing $2^2 = 4$ to the value of the summation. Thus, if plane $n = 1$ contains h_1 non-zero bits, $D_1 = 4w_B^2 h_1$. Generalizing this, we can rewrite (1) as

$$D_n = w_B^2 2^{2n} h_n, \tag{3}$$

where h_n is the number of non-zero bits in the plane n.

In EBCOT, the fractional bit planes for an image are sorted according to their distortion metric and their length. For a lossy image, the bit planes with the least quality contribution per bit are discarded, that is, rate distortion optimization methods are used to sort the bit planes according to their bit rate contribution and to their distortion contribution. For video coding, this process should not be done per frame. Suppose that a particular bit plane of significance n is dropped from a frame sub-band, and this frame is the reference frame used to predict the current frame. The prediction will have an error of significance n because of the dropped bit plane in the reference frame, so if the residual is decoded for the current frame, it will be added to an inaccurate bit plane, and will result in no improvement in the frame quality. Thus, if a bit plane is dropped in a frame, it should be dropped in the subsequent frames in the group of pictures.

To deal with this issue, the length and the quality contribution of each bit plane for all the frames in the group are added together, giving one total bit plane length and one total quality contribution for each bit plane in all frames across the group of pictures. The bit planes are then sorted using these total values. Since the frames in a group of picture are typically of a similar nature, this should not have a large effect on the rate distortion optimization.

A list of the bit planes in order is then generated. When selecting a lower-quality subset from a sequence, the bit planes at the end of the list are discarded first. If to obtain the desired bit rate a bit plane is to be discarded, it is discarded from all the frames in the group of pictures. If the total length of the bit plane across all the frames is larger than the length that needs to be truncated for the desired bit rate to be obtained, then the bit plane does not need to be discarded from all the frames; it may be retained in a number of frames at the beginning of the group of pictures, and discarded in the rest.

4 EXPERIMENTAL RESULTS

The proposed scalable video coding method was tested using the first 10 frames of the *pedestrian area* video sequence [21] with a resolution of 1920×1080 and a frame rate of 25 fps. Tests on other video sequences showed similar behaviour, so only the results for the *pedestrian area* sequence are shown in this paper. The rate distortion characteristics were compared to those using the H.264/AVC JM reference software version 17. with the main profile [13]. To make the comparison fair, bidirectional frames and sub-pixel motion estimation where disabled from the JM encoder, and only one reference frame was used.

The three motion estimation methods mentioned in Section 3.1.1 were compared, and the results can be seen in Fig. 5. They are also compared to the H.264/AVC JM reference encoder. The performance of the three estimation methods is

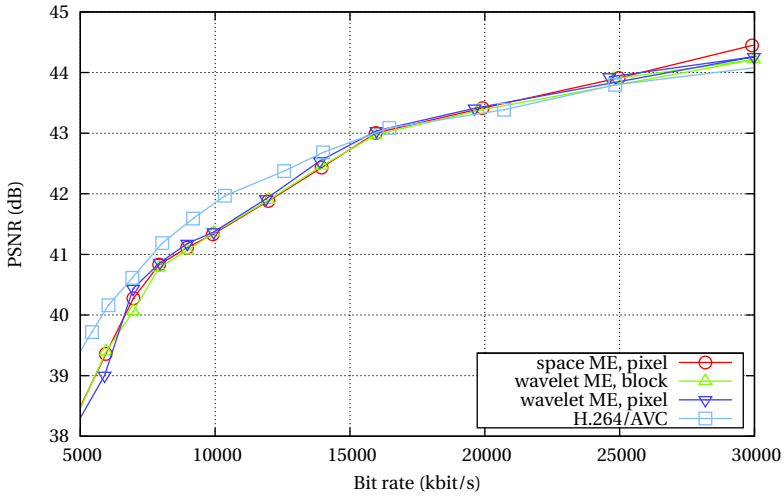

Fig. 5: Rate distortion characteristics for three different motion estimation methods with three levels of decomposition and the 9/7 filter, as well as for H.264/AVC using the JM reference encoder.

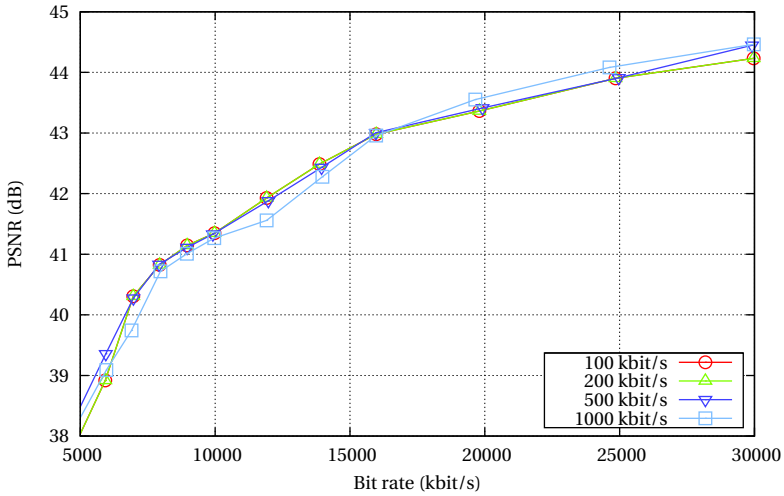

Fig. 6: Rate distortion characteristics for motion estimation in space domain with three levels of decomposition and the 9/7 filter for different base layer bit rates.

very close, so the motion estimation in space domain was selected for the other experiments. Using motion estimation in space domain would make it easier to use the same motion estimation engine as that for [17] in a reconfigurable universal compression system.

In Fig. 5, the curves for the proposed video coding scheme have rate distortion characteristics which are reasonably close to those of H.264/AVC. Moreover, it must be mentioned that for each point in the H.264/AVC curve, the JM encoder had to encode the video sequence from scratch, whereas for each of the other curves, the video sequence was encoded only once, and then a subset taken for each point without the need to re-encode the sequence. Also, when the bit rate goes up and start to approach lossless quality, the compression performance of the proposed video coding scheme is better than that obtained by the JM encoder. For very low bit rates, the JM encoder performs better than the proposed method.

Fig. 6 shows a comparison of using different base layer bit rates. Using a bit rate of 500 kbit/s was found to be suitable across the different motion estimation methods, and also for different numbers of levels of decomposition and different wavelet filters. For different frame resolutions, a different base layer bit rate would need to be selected.

Finally, Fig. 7 shows the performance of the encoder for different numbers of levels of decomposition and different wavelet filters. The 9/7 filter gives better results than the 5/3 filter in the range of the shown plot. It should be noted that the video sequence cannot be decoded losslessly when using the 9/7 filter. If it is important for a lossless version of the image to be available, the 5/3 filter has to be used instead.

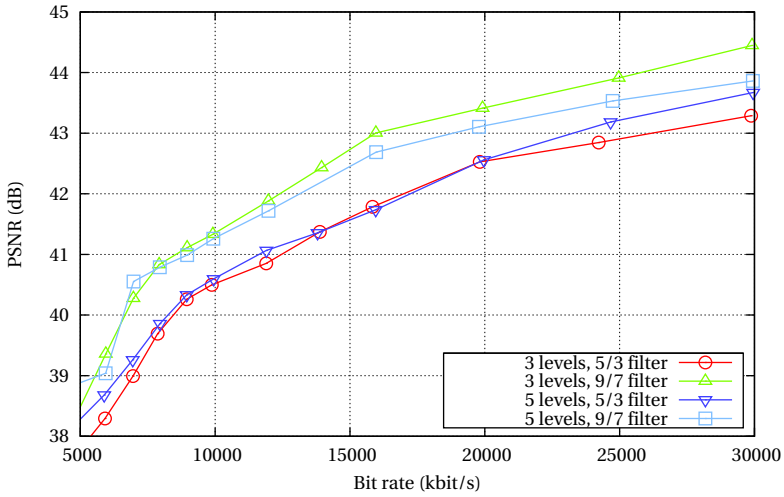

Fig. 7: Rate distortion characteristics for motion estimation in space domain with a base layer bit rate of 500 kbit/s for different decomposition levels and filters.

5 CONCLUSION

In this paper, a fine-grained scalable video coding method inspired by the EBCOT and ICER image encoders was presented. This method can be combined with other compression methods in a universal hardware compression system. Although motion compensation is used by the scheme, no explicit motion vectors need to be transmitted, as they can be regenerated by the decoder. The scheme allows the quality of the encoded bitstream to be degraded by simply transmitting a subset of the whole bitstream, with no extra video coding required. Experimental results show that the rate distortion characteristics of the scalable coding scheme are reasonably close to H.264/AVC with the added advantage of scalability.

References

[1] G. Yu, T. Vladimirova, and M. N. Sweeting, "Image compression systems on board satellites," *Acta Astronautica*, vol. 64, no. 9–10, pp. 988–1005, Feb. 2009.

[2] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[3] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[4] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[5] 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, Jun. 1996.

[6] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[7] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 36–58, Sep. 2001.

[8] A. Kiely and M. Klimesh, "The ICER progressive wavelet image compressor," Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, The Interplanetary Network Progress Report 42-155, Jul.–Sep. 2003. [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report/42-155/155J.pdf

[9] D. Taubman, "Successive refinement of video: Fundamental issues, past efforts and new directions," in *International Symposium on Visual Communications and Image Processing*, vol. 5150. SPIE, 2003, pp. 649–663.

[10] A. Secker and D. Taubman, "Highly scalable video compression with scalable motion coding," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1029–1041, Aug. 2004.

[11] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[12] SVC reference software (JSVM software). [Online]. Available: http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm

[13] H.264/AVC JM reference software. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[14] J. L. Nuñez-Yañez, X. Chen, N. Canagarajah, and R. Vitulli, "Statistical lossless compression of space imagery and general data in a reconfigurable architecture," in *NASA/ESA Conference on Adaptive Hardware and Systems, 2008*, Jun. 2008, pp. 172–177.

[15] J. L. Nuñez-Yañez and V. A. Chouliaras, "A configurable statistical lossless compression core based on variable order Markov modeling and arithmetic coding," *IEEE Transactions on Computers*, vol. 54, no. 11, pp. 1345–1359, Nov. 2005.

[16] X. Chen, N. Canagarajah, J. L. Nuñez-Yañez, and R. Vitulli, "Lossless compression for space imagery in a dynamically reconfigurable architecture," in *Workshop in Applied Reconfigurable Computing*, 2008, pp. 336–341.

[17] X. Chen, N. Canagarajah, and J. L. Nuñez-Yañez, "Backward adaptive pixel-based fast predictive motion estimation," *IEEE Signal Processing Letters*, vol. 16, no. 5, pp. 370–373, May 2009.

[18] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, Apr. 1988, pp. 761–764.

[19] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[20] Y. Liu and K. Ngi Ngan, "Fast multiresolution motion estimation algorithms for wavelet-based scalable video coding," *Signal Processing: Image Communication*, vol. 22, no. 5, pp. 448–465, Jun. 2007.

[21] Lehrstuhl für Datenverarbeitung, Technische Universität München. Test sequences 1080p. [Online]. Available: ftp://ftp.ldv.e-technik.tu-muenchen.de/dist/test_sequences/1080p/