SCALABLE MOTION VECTOR CODING

J. Barbarien[*]*, A. Munteanu, F. Verdicchio, Y. Andreopoulos, J. Cornelis and P. Schelkens*

Vrije Universiteit Brussel
Department of Electronics and Information Processing
Pleinlaan 2, B-1050 Brussels, Belgium
[*]jbarbari@etro.vub.ac.be

ABSTRACT

Recently proposed scalable wavelet-based video codecs using spatial-domain motion compensated temporal filtering (SDMCTF) offer competitive compression performance when compared to H.264 and generate embedded bit-streams supporting quality, resolution and temporal scalability. To be able to support a large range of bit-rates with optimal compression efficiency, these codecs require a quality-scalable motion vector coding technique. Such an algorithm based on the integer wavelet transform followed by embedded coding of the wavelet coefficients was proposed in the recent past. In this paper, we present a quality-scalable motion vector coding algorithm using median-based motion vector prediction. The compression performance of the proposed algorithm is compared to that of the wavelet-based technique and is found to be superior. Additionally, the proposed motion vector codec is incorporated into an SDMCTF-based video codec and the benefits of using quality-scalable motion vector representations are experimentally demonstrated.

1. INTRODUCTION

In the past, it was always deemed necessary to encode the motion information generated by scalable video codecs in a lossless fashion to obtain acceptable compression efficiency. A first drawback of this approach is that the minimum attainable bit-rate for the video codec is bound by the rate needed to losslessly code the motion information. As a consequence, motion estimation efficiency must often be sacrificed when very low bit-rates need to be supported. This lowers the compression performance at all bit-rates. A second drawback is that, at low rates, most of the available rate is spent on motion data, while a better distribution of the rate between motion data and texture data may deliver a better output quality. To solve these problems, a quality scalable motion vector representation must be introduced [1]. Quality scalable motion vector coding techniques that perform an integer wavelet transform of the motion vector components followed by embedded coding of the resulting wavelet coefficients were proposed in [2] and later in [1]. The compression efficiency of this kind of coding schemes is however significantly lower than that of traditional non-scalable motion vector codecs based on motion vector prediction [2]. In this paper, a new motion vector codec is introduced, combining high compression efficiency obtained by using median-based motion vector prediction with support for quality scalability. The details of the proposed motion vector codec are presented in section 2. The conducted experiments and their results are discussed in section 3. Finally, the conclusions of the paper are summarized in section 4.

2. QUALITY-SCALABLE PREDICTION-BASED MOTION VECTOR CODING

2.1. Structure of the motion information

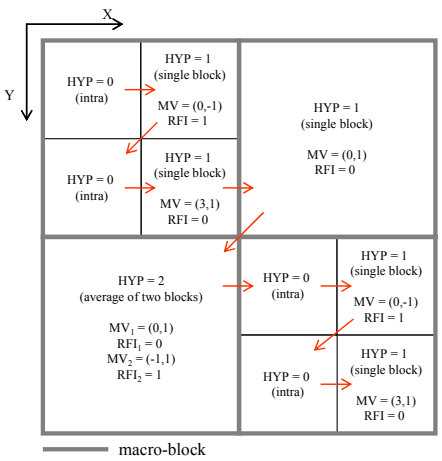

Figure 1: Example of the motion information generated by multi-hypothesis block-based motion estimation using multiple block sizes and reference frames.

The proposed motion vector codec (MVC) is designed to compress motion information produced by multi-hypothesis block-based motion estimation (ME) using multiple block sizes and multiple reference frames [3],[4]. Figure 1 shows an example of the information produced by such a ME algorithm. For each macro-block, the motion information that needs to be encoded consists of:

1. If and how the macro-block is split into sub-blocks. This will be referred to as the splitting information.
2. For each separately predicted block:

a. The hypothesis, meaning the way the block is predicted (intra, by a single block in one of the reference frames or by the average of multiple blocks each lying in one of the reference frames).

b. Depending on the hypothesis, zero (intra), one or more motion vectors and for each vector the associated reference frame index.

The next sections will describe the way this information is coded in the proposed MVC.

2.2. General architecture of the MVC

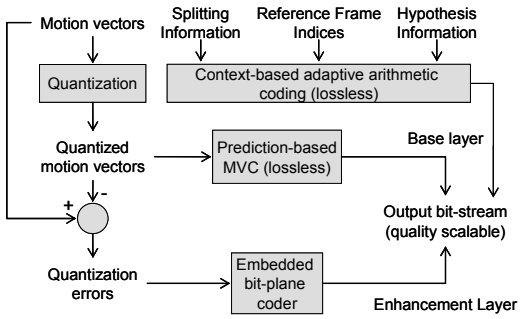

Figure 2: General setup of the proposed MVC.

The presented MVC attempts to combine the compression efficiency of prediction-based MVCs with quality scalability. The most straightforward approach to achieve this would be to combine motion vector prediction with quality scalable coding of the prediction errors. This is however not a viable solution. Because of the closed-loop nature of the prediction, errors produced by lossy decoding can cascade, causing severely degraded and unusable motion vector data to be decoded. To solve this drift phenomenon, a new motion vector coding architecture is introduced (see Figure 2). The proposed algorithm generates a bit-stream consisting of a base layer, which must always be decoded losslessly, and an enhancement layer that is quality scalable.

The splitting information, reference frame indices and hypothesis information are all coded using context-based adaptive arithmetic coding. The encoded information becomes a part of the base layer of the final bit-stream. The motion vectors themselves are first quantized by discarding the information on the lowest bit-plane(s). The quantized motion vectors are thereafter compressed using the prediction-based motion vector coding technique described in section 2.3. The resulting compressed data is added to the base-layer part of the final bit-stream. The quantization errors are coded using the embedded bit-plane coding algorithm discussed in section 2.4. This data forms the quality scalable enhancement layer of the final bit-stream. In this way, the total motion vector bit-rate can be chosen to lie anywhere between the bit-rate needed to code the base-layer losslessly and the bit-rate needed for a complete, lossless reconstruction of the motion information. The base layer bit-rate can be controlled in the encoder by changing the quantization step size. Using a smaller base layer size increases the range of supported bit-rates but in most cases decreases the overall compression performance of the MVC (see section 3).

2.3. Prediction-based coding of the quantized motion vectors

The quantized motion vectors are encoded by performing motion vector prediction followed by lossless coding of the resulting prediction errors. Both in the prediction step and in the prediction-error coding step, the macro-blocks are visited in raster order. For a split macro-block, the sub-blocks are visited in depth-first quadtree scanning order. The employed scanning order is illustrated with red arrows in Figure 1. The following notations are adopted:

- W and H : Width and height of the predicted frame.
- B_c : The currently visited block. The size of B_c is $S_c \times S_c$ pixels. The top-left pixel of block B_c has coordinates (x_c, y_c) .
- N_c : The number of different motion vectors generated in the ME process to predict B_c .
- MV_i^c : If $N_c > 0$, the i -th, motion vector involved in the prediction of B_c , with $1 \le i \le N_c$.
- RFI_i^c : The reference frame index associated with MV_i^c .

The motion vector prediction is initiated by constructing a pixel-based motion field P . This is done by associating with each pixel in the predicted frame the motion information of the smallest block the pixel belongs to. The coordinate system depicted in Figure 1 is used. The following notations are adopted:

- $N(x,y)$: The number of different motion vectors generated in the ME process for the prediction of the smallest block containing the pixel at position (x,y) .
- $MV_i(x,y)$: If $N(x,y) > 0$, the i -th, motion vector generated in the ME process to predict the smallest block containing the pixel (x,y) , $1 \le i \le N(x,y)$.

The prediction of $MV_j^c, 1 \le j \le N_c$ is performed by taking the median of a set of quantized motion vectors U_p . The following vectors are considered for insertion into U_p :

1. $MV_i(x_c - 1, y_c + k), 1 \le i \le N(x_c - 1, y_c + k), 0 \le k < S_c$ if $0 \le x_c - 1 < W$ and $0 \le y_c + k < H$ (motion vectors of the pixels located directly to the left of the currently visited block).
2. $MV_i(x_c + k, y_c - 1), 1 \le i \le N(x_c + k, y_c - 1), 0 \le k < S_c$ if $0 \le x_c + k < W$ and $0 \le y_c - 1 < H$ (motion vectors of the pixels directly above the currently visited block).

3. $MV_i\left(x_c+S_c, y_c-\frac{S_c}{2}+k\right), 1\le i\le N\left(x_c+S_c, y_c-\frac{S_c}{2}+k\right), 0\le k<\frac{S_c}{2}$ if $0\le x_c+S_c<W$ and $0\le y_c-\frac{S_c}{2}+k<H$ and $MV_i(x_c+S_c+k, y_c-1), 1\le i\le N(x_c+S_c+k, y_c-1), 0\le k<\frac{S_c}{2}$ if $0\le x_c+S_c+k<W$ and $0\le y_c-1<H$ (motion vectors of the pixels to the top-left of the currently visited block).
4. $MV_i^c, 1\le i<j$ (previously-predicted quantized motion vectors belonging to the currently visited block). These vectors are each added S_c times to the set to have the same impact on the median as the spatially neighbouring motion vectors.

From the quantized motion vectors in the four sets above, only the ones that (a) were predicted earlier in the motion vector prediction procedure (causality), and (b) have a reference frame index pointing to a frame with the same distance (in number of frames) to the predicted frame as the frame pointed to by RFI_j^c, are added to U_p. If the reference frame index of a considered motion vector points to a future frame in the sequence, the signs of the components are inverted prior to insertion into U_p.

Context-based adaptive arithmetic coding is used to encode the prediction errors. The horizontal and vertical components of the prediction errors are coded separately. The interval of possible component values is split into a number of sub-intervals:

$$S_i=\begin{cases}\{0\}\ if\ i=0\\ \left[-\left(2^i-1\right),-\left(2^{i-1}\right)\right]\cup\left[2^{i-1},2^i-1\right]\ if\ i>0\end{cases} \quad (1)$$

For each component value, a symbol representing the sub-interval it belongs to is coded first. Thereafter, the offset of the prediction error component within the sub-interval is coded. One model of the arithmetic coder is used to code the sub-interval index. For each interval, a different model is used to encode the offset values. For sub-intervals containing less then 16 values, the offset is directly coded as a symbol of the model. In the other case the model only allows two symbols 0 and 1, and the offset is coded in its binary representation.

2.4. Embedded coding of the quantization errors

The horizontal and vertical components of the quantization errors are coded in a bit-plane by bit-plane fashion. Each bit-plane is coded in two passes, a refinement pass followed by a significance pass. With each bit-plane i, a threshold $T=2^i$ is associated. A quantization error component c_e is said to be significant for a threshold T if $|c_e|\ge T$. All bit-planes are coded sequentially, starting with the highest bit-plane. In the significance pass, the significance of all previously non-significant components is encoded. When a component becomes significant for the first time and its corresponding quantized motion vector component is 0, its sign is also coded. The macro-blocks are visited in raster order. For a split macro-block, the sub-blocks are visited in depth-first quadtree scanning order. Multiple quantization error vectors associated to the same block are visited sequentially before proceeding to the next block. In the refinement pass, already significant components are refined by coding the binary value on the current bit-plane of the component. The significance, refinement and sign information is compressed using context-based adaptive arithmetic coding. Denote by $QE(MV)$ the quantization error associated with quantized motion vector MV. The significance information for the components of $QE\left(MV_1^c\right)$ is coded using three probability models per component. The choice of the model is based upon the significance of the corresponding components of $QE\left(MV_1(x_c-1, y_c)\right)$ and $QE\left(MV_1(x_c, y_c-1)\right)$. Model 1 is used if both components are significant or if one of the components is significant and the state of the other is unavailable (for instance if $QE\left(MV_1(x_c, y_c-1)\right)$ does not exist). The second model is used if both components are not significant or if one of the components is not significant and the state of the other is unavailable. The third model is used in all other cases. Finally, two models for each component are used to code the significance of the components of $QE\left(MV_i^c\right)$ for all $i, 1<i\le N_c$. The first model is used if the corresponding component of $QE\left(MV_1^c\right)$ is significant, the second if it is not. Only one model is used to code the refinement information for both components and one model per component is reserved to code the sign information.

3. EXPERIMENTAL RESULTS

In a first experiment, the lossless compression performance of the proposed MVC is compared to that of a wavelet-based quality-scalable motion vector coding technique, as proposed in [1], [2]. The motion information used in this comparison is generated by a multi-hypothesis motion estimation algorithm, using two reference frames, no intra mode, no macro-block splitting and quarter-pel accuracy, embedded in an SDMCTF codec framework [5]. This type of motion data is chosen to avoid difficulties in adapting the wavelet-based MVC to more complex motion information. The employed wavelet-based quality scalable MVC separately codes the components of the motion vectors by performing a 5/3 integer wavelet transform followed by quality scalable coding of the resulting transform coefficients using the QT-L codec of [6]. The performance of this codec is on par with JPEG2000 [6].

Name	Resolution	Nr. of frames	Framerate (Hz)
Football	CIF	260	30
Canoa	CIF	220	30
Container	CIF	300	30

Table 1: Test sequences used in our experiments.

The experiments were conducted for the three sequences described in Table 1. In Table 2, the average number of bytes spent on the motion vectors of a predicted frame is shown for the wavelet-based MVC and for the proposed MVC using different quantization steps ($Q = i$ means the lowest i bit-planes are discarded in the motion vector quantization).

Sequence	Proposed MVC					Wavelet-based MVC
	Q=0	Q=1	Q=2	Q=3	Q=4	
Football	652	662	673	682	689	722
Canoa	642	650	656	664	677	709
Container	161	162	158	154	150	211

Table 2: Comparison between the proposed MVC and a wavelet-based MVC for different quantization step sizes.

In the second experiment, the proposed MVC is incorporated into an SDMCTF-based video codec using multi-hypothesis motion estimation with two reference frames, two block sizes and quarter-pel accuracy [3]. The overall compression performance of the video codec is compared when using scalable and non-scalable motion vector coding (the later corresponding to the proposed MVC with no quantization and no enhancement layer). The base layer size in the scalable MVC is kept below 96 kbps by appropriately selecting the quantization step size for each predicted frame. The distribution of the rate between texture data and motion information is done using a heuristic technique similar to the one described in [1]. For each target bit-rate, the motion vectors are decoded at 6 different rates, evenly spread out between the base-layer rate and the rate needed for the lossless reconstruction of the vectors. The remaining bit-rate is spent on the texture coding. The combination of motion and texture rates delivering the best quality (PSNR) is retained. The results of the experiment are shown in Table 3 for the "Football" and "Canoa" sequences described in Table 1. We report the average PSNR (in dB) per frame when decoding to different target rates.

4. CONCLUSIONS

In this paper, we introduce a drift-free quality-scalable prediction-based motion vector coding technique. Our first experiment shows that the proposed MVC outperforms wavelet-based quality-scalable motion vector coding [1], [2]. The second experiment highlights the benefits of integrating the designed quality-scalable MVC into an SDMCTF-based video coding framework. Lower bit-rates can be attained without sacrificing motion estimation efficiency and overall coding performance at low rates is improved by better distribution of the available rate between texture and motion information.

	Target rate (kbps)	128	192	256	512	1024
Football	Non-scalable MVC	-	22.99	25.99	28.24	30.84
	Scalable MVC	21.76	24.98	26.04	28.23	30.83
Canoa	Non-scalable MVC	-	-	23.16	26.10	28.58
	Scalable MVC	20.16	22.30	23.55	26.05	28.55

Table 3: Comparison between scalable and non-scalable motion vector coding.

5. ACKNOWLEDGEMENTS

This work was supported by IWT (PhD bursary Joeri Barbarien) and DWTC (IAP Phase V - Mobile Multimedia). P. Schelkens has a post-doctoral fellowship with the Fund for Scientific Research − Flanders (FWO), Egmontstraat 5, B-1000 Brussels, Belgium.

6. REFERENCES

[1] D. Taubman and A. Secker, "Highly scalable video compression with scalable motion coding," Int. Conf. Image Processing (ICIP 2003), Sept. 2003.

[2] J. Barbarien, Y. Andreopoulos, A. Munteanu, P. Schelkens and J. Cornelis, "Coding of motion vectors produced by wavelet-domain motion estimation," ISO/IEC JTC1/SC29/WG11, m9249, Awaji Island, Japan, Dec. 2002.

[3] Y. Andreopoulos, J. Barbarien, F. Verdicchio, A. Munteanu, M. van der Schaar, J. Cornelis, P. Schelkens, "Response to the call for evidence on scalable video coding advances," ISO/IEC JTC1/SC29/WG11, m9911, Trondheim, Norway, July 2003.

[4] M. Flierl, T. Wiegand, and B. Girod, "Rate-constrained multihypothesis prediction for motion-compensated video compression," IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 11, pp. 957-969, Nov. 2002.

[5] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis, "Open-loop, in-band, motion-compensated temporal filtering for objective full-scalability in wavelet video coding," ISO/IEC JTC1/SC29/WG11, m9026, Shanghai, China, Oct. 2002.

[6] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro I Nieto, and J. Cornelis, "Wavelet Coding of Volumetric Medical Datasets," IEEE Transactions on Medical Imaging, vol. 22, no. 3, pp. 441-458, March 2003.