New flexible motion estimation technique for scalable MPEG encoding using display frame order and multi-temporal references

NEW FLEXIBLE MOTION ESTIMATION TECHNIQUE FOR SCALABLE MPEG ENCODING USING DISPLAY FRAME ORDER AND MULTI-TEMPORAL REFERENCES

Stephan Mietens (1), Gerben Hekstra (3), Peter H.N. de With (1,2), and Christian Hentschel (3)

Eindhoven Univ. of Technology - EESI(1) / CMG Eindhoven(2), P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Philips Research Labs.(3), Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

ABSTRACT

The applicability of MPEG video coding can be improved by scaling both the algorithmic complexity and resource usage appropriately for the intended device and application. For this purpose, we present a new technique for motion estimation, based on a scalable three-stage process including frame processing in display order, approximation of motion vector fields using multiple references and optional quality refinements. Experiments show that the computational effort is scalable with a factor of 14, resulting in a global variation of 7 dB SNR in picture quality. At full processing, our technique slightly outperforms a 32×32 full search motion estimation. The technique forms a valuable contribution to mobile MPEG coding applications, following the scalability concepts introduced in [1].

1. INTRODUCTION

Internet-based applications (e.g. video conferences), portable television applications, and mobile consumer terminals impose variable video quality requirements on the system architecture. These requirements can be exploited for reduction of the algorithmic complexity of applications such as MPEG coding, while accepting a certain quality loss under circumstances as indicated below. Firstly, a part of the available general-purpose computation power of a TV can be saved to perform other tasks in parallel. Secondly, when using small displays in e.g. mobile devices, the observer cannot perceive fine details that are contained in the complete video signal. However, the corresponding applications still perform a full and thus costly processing of the signal.

It is our objective to design a scalable MPEG encoding system that features scalable video quality and a corresponding scalable resource usage [2]. Such a system enables advanced video encoding applications on a plurality of low-cost or mobile consumer terminals having limited resources (available memory, bandwidth, computing power, stand-by time, etc.) compared to high-end computer and TV systems.

An expensive part and corner stone in the signal processing complexity of an MPEG encoder is motion estimation (ME). The search of motion vectors requires significant computation power, because large temporal distances between reference frames lead to large search ares. Furthermore, video frames in a group-of-pictures (GOP) are processed in a reordered way, e.g. "IPBB" (transmit order) in place of "IBBP" (display order), which forecloses the reuse of intermediate results for better motion prediction. Instead, we perform an initial motion estimation with the video frames at the entrance of the encoder (thus in display order) and found considerable savings in computation and advantages for scalibility. The initial estimation is exploited to efficiently derive the desired motion vector fields needed for the final MPEG encoding process. Furthermore, the quality of full-search motion estimation can be obtained with an optional refinement stage. Figure 1 shows the principle architecture of the aforementioned tasks.

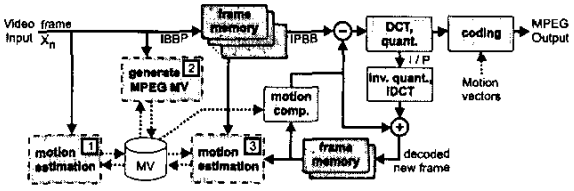

Fig. 1. Simplified architecture of the new three-stage ME.

The paper is organised as follows. The problem statement is introduced in Section 2. Section 3 presents a new three-stage method to perform the ME with considerable savings in computing power and memory bandwidth for resource-constrained applications. Section 4 shows experimental results and Section 5 concludes the paper.

Notations

For the ease of discussion, we form subgroups of pictures (SGOP) that have the form $(I/P)BB...B(I/P)$ within a group of pictures (GOP) defined in MPEG and we refer to Figure 2.

The number of pictures within a subgroup k is denoted by M_k, analogous to the prediction depth M of a GOP, and can vary from SGOP to SGOP. The MPEG forward vector-field, which is used in the prediction of the i^{th} frame, is denoted by f_i^k. The MPEG backward vector-field is denoted

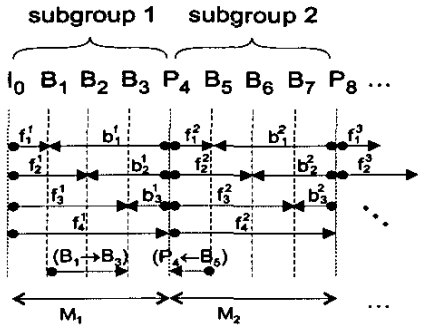

Fig. 2. Example of vector fields used for motion estimation in MPEG encoding after defining a GOP structure. In this example, a GOP with a constant $M = 4$ was chosen.

by b_i^k. Arbitrary vector fields are denoted by $(X_m \rightarrow X_n)$ for the forward case and $(X_m \leftarrow X_n)$ for the backward case, indicating motion between frame X_m and X_n with $n > m$. To indicate the frame type of X, it can be replaced by I, P, or B. In this paper, we discuss ME in terms of entire vector fields, rather than the individual motion vectors. Computations in this paper act on entire vector fields.

2. PROBLEM STATEMENT

A large number of ME algorithms has been proposed for reducing the computation complexity of a full search. The algorithms make a trade-off between the complexity and quality of the computed vector fields. When compared to full search, popular algorithms like New Three Step Search [3] and Center-Biased Diamond Search [4] provide a good quality of motion vector fields at low cost. However, the accuracy of the motion vectors is limited for fast motion in the video sequence.

A further reduction of the computation complexity is achieved by using recursive motion estimation (RME, already discussed in [5][6]) that derives candidate motion vectors from previously computed motion vectors in both the current motion vector field (so-called "spatial" candidates) or the previous motion vector field (so-called "temporal" candidates). Up to now, the usage of the RME algorithms has been limited to a GOP structure with fixed M, and only $(I \rightarrow P)$ vector fields f_M^k are used once for the computation of the next SGOP fields f_M^{k+1} without modification.

The problem of the aforementioned algorithms for ME in MPEG is that a larger value of M increases the prediction depth, which implies a larger frame distance between reference frames, thereby making it difficult to accurately estimate the motion. To overcome this problem, we introduce a new three-stage ME, featuring display-order frame processing and a concept called *multi-temporal* ME in the next Section. These new techniques give more flexibility for a scalable MPEG encoding process.

3. THREE-STAGE DISPLAY-ORDER MULTI-TEMPORAL MOTION ESTIMATION

Temporal candiate motion vectors used in RME give good predictions of the current motion in a video sequence. For this reason we will use RME based on block matching within our new architecture for ME, and it will be applied for the computation of all vector fields.

It is obvious that the prediction quality improves with a smaller temporal distance D. The parameter D denotes the difference between the frame numbers of the considered frames. The conventional computation of vector fields is going from I- to P- and then the B-frames in between. The first corresponding vector fields from I to P has a temporal distance $D = M$, and the forward and backward vector fields for the B-frames have distances $D < M$. When abstracting from this computation order, the prediction of an SGOP requires nothing else but a set of vector fields having various temporal distances $0 < |D| \leq M$. The required vector fields can be predicted themselves from other vector fields and the quality of this prediction improves with decreasing D. This has two implications. First, when using RME, it can be easily seen that the computation order C of the vector fields in a SGOP k should be $C_{opt} = \{f_1^k, f_2^k, f_3^k, f_4^k, b_3^k, b_2^k, b_1^k\}$. Second, a forward or backward predicted vector field $F_1 \in \{f, b\}$ that is used as a temporal candidate for another motion vector field $F_2 \in \{f, b\}$ should be scaled to an appropriate temporal distance as specified by

$$F_2 = \frac{d(F_2)}{d(F_1)} * F_1, \qquad (1)$$

where $d(F)$ is a function that returns the temporal distance D that is covered by the given vector field F. Note that $D > 0$ for f and $D < 0$ for b.

To indicate the difference in perfomance, we have compared the quality resulting from C_{opt} with the quality resulting from the order $C_{MPEG} = \{f_1^k, b_1^k, f_2^k, b_2^k, f_3^k, b_3^k, f_4^k\}$, which defines the MPEG standard processing order. Figure 3 shows that C_{opt} results in up to 5.5 dB higher signal-to-noise ratio (SNR) for the reconstructed frames in case of motion than using C_{MPEG} with the same RME. Note that when compared to a 32×32 full search, the average SNR of the reconstructed frames using C_{opt} is 0.1 dB lower for B-frames and 1.46 dB *higher* for P-frames.

This preliminary experiment leads to the desire to reduce the temporal distance to the minimum $|D| = 1$, given the advantage of obtaining accurate ME and a low computation cost due to the minimum temporal distance. However, this cannot directly be implemented because the MPEG GOP structure has $M > 1$ and thus some of the required vector fields have $D > 1$. For this reason, we split up the ME into stages. The first stage aims at deriving *prediction of vector fields* and for maximum performance combined with simplicity, we use RME with only $|D| = 1$. In a second

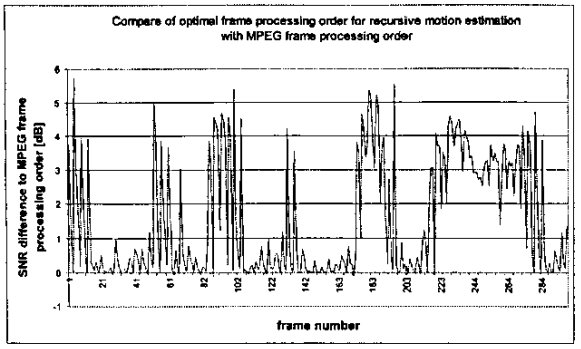

Fig. 3. SNR difference of reconstructed frames ($N = 16$, $M = 4$) of the "Stefan" sequence (tennis scene) using either computation order C_{opt} or C_{MPEG}. C_{opt} results in up to 5.5 dB higher SNR than C_{MPEG}.

stage, these predicted vector fields are used to calculate the required vector fields according to the MPEG standard. In a further stage, the vector fields can be refined by an additional - although simple - ME. This new concept in our system results in a three-stage process as follows.

- *Stage 1.* Prior to defining the GOP structure, we perform a simple RME for every frame X_n and compute the forward motion-vector field $(X_{n-1} \rightarrow X_n)$ and then the backward field $(X_{n-1} \leftarrow X_n)$. For example, in Figure 2 this means computing vectors like f_1^1 and b_3^1, but then for every pair of sequential frames.
- *Stage 2.* After defining a GOP structure, all vector fields $F \in \{f, b\}$ required for MPEG encoding are approximated using a new concept called *multi-temporal*ME, which is introduced below (a variant of this with the same name is used in H.26L coding). The approximation of the fields F is performed by appropriately accessing multiple available vector fields F_A and F_B and combine them using the linear relation

$$F = \alpha * F_A + \beta * F_B, \qquad (2)$$

where the scaling factors α and β depend on the processed fields to obtain the correct temporal distance needed for F. For example, $f_2^1 = f_1^1 + (B_1 \rightarrow B_2)$ (see Figure 2), thus having $\alpha = \beta = 1$. Note that α and β become different if the frame distances change or when complexity scaling is applied (see below). The term multi-temporal refers to two aspects. Firstly, the computation of Equation (2) means that one vector field is combined from two other vector fields. Secondly, the total prediction of a vector field can be based on various vector fields such that several temporal references are used. The second aspect can be used for high-quality applications to approach different motions like real velocity, zoom, etc.

- *Stage 3.* For final MPEG ME in the encoder, the computed approximated vector fields from the previous stage are used as an input. Beforehand, an optional refinement of the approximations can be performed with a second iteration of RME.

Architecture

The aforementioned three stages can be found back in Figure 1. In this Figure, it can be seen that the three stages are interconnected via a memory for motion vectors. Therefore, each stage can take advantage from the avaibility of several motion vector fields that are computed by all stages. This advantage can be used to enhance vector field predictions. Furthermore it can be used to reduce the computational effort of the motion estimation processes in Stage 1 and 3. Note that the amount of memory needed for the new architecture is the same as used for a standard encoder, except for the memory needed for motion vectors. The additional memory (2 bytes per vector) is negligible, compared to the memory consumption of a video frame (256 bytes per macroblock for luminance only).

The principal advantage of the presented flexible framework is that it provides an excellent starting position for quality and computation scaling in the MPEG encoding process, as illustrated below.

- Stage 1 can omit the computation of vector fields (e.g. the backward vector fields) or compute only significant parts of a vector field to reduce the computational effort and memory.
- The effort for computing a vector field can be reduced in all stages if a constant motion velocity is measured in the current video stream.
- The set of motion vectors that are refined in Stage 3 can be limited to significant parts of a vector field to save computations and memory accesses. If this refinement is omitted completely, the new technique can take advantage of further reduced computational effort, because the processing of the vector fields in Stage 1 and 2 is much simpler than ME itself when applied to frames with a large temporal distance.

An alternative aspect of the architecture is that the three stages are nearly fully decoupled from the actual coding process, thereby giving high flexibility for parallelisation. Furthermore, the achitecure provides original frames without quantisation errors to Stage 1 (this performs the initial RME process on succeeding frames). For these reasons, the RME results in accurate motion vector fields.

4. EXPERIMENTS AND RESULTS

The flexible scalable motion-estimation technique we introduced in Section 3 gives a good opportunity for quality and computational scaling. The results of a proof-of-concept

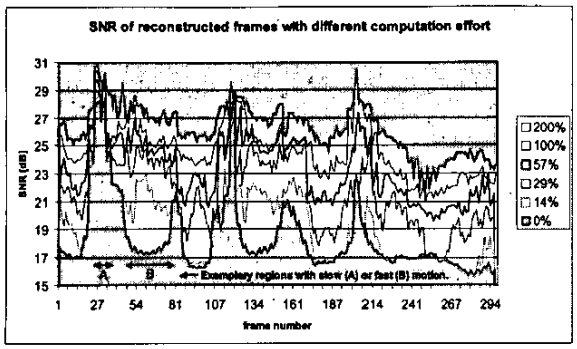

Fig. 4. SNR of reconstructed B-frames of the "Stefan" sequence (tennis scene) with different computational effort, P-frames are not shown for the sake of clarity ($N = 16$, $M = 4$).

experiment using the "Stefan" sequence (tennis scene) is shown in Figure 4, based on a GOP size of $N = 16$ and $M = 4$ (thus "IBBBP" structure). We use the RME taken from [5] (limited to pixel-search) in Stage 1 and 3 because of its simple design. Note that the RME for this proof-of-concept does not yet take advantage of the reduced temporal distance during the initial stage at the entrance of the encoder (e.g. by testing less vector candidates). The area with the white background shows the scalability of the quality range that results from downscaling the amount of computed motion-vector fields. Each vector field requires 14% of the effort compared to a 100% simple RME based on 4 forward vector fields and 3 backward vector fields when going from one to the next reference frame. If all vector fields are computed and the refinement stage 3 is performed, the computational effort is 200% (not optimised). In this experiment, the order of including vector-field computations to increase the complexity level was simply f_1, f_2, f_3, f_4, b_3, b_2, b_1 for Stage 1 and then the same order was used for Stage 3 to refine these vector fields. The results of the scalability in quality and computational effort in our experiment are as follows. The average SNR of our ME technique of the reconstructed P- and B-frames prior computing the differential signal and quantisation are 25.16 dB, 24.55 dB, 23.52 dB, 22.48 dB and 18.58 dB for 200%, 157%, 100%, 57% and 14%, respectively. For a full quality comparison (200%), we consider full-search block-matching. A full-search ME with a search window of 32×32 pixels results in an average SNR of 24.80 dB. Thus our new technique slightly outperforms full search by 0.36 dB for this test sequence.

Note that since it is not needed to define a GOP structure prior to the first stage of the new ME method, the GOP structures can be dynamically adapted based on e.g. an analysis of the computed vector fields. Furthermore, such an analysis can be used to decide whether to skip the computation of specific fields, e.g. if no motion was detected in the previous frames. This reduces also the memory bandwidth usage, because frames are not accessed in this case.

Furthermore note that a state-of-the-art ME algorithm itself can be improved using multi-temporal vector field predictions. This implies that a higher number of predictions are generated for the computation of one vector field.

5. CONCLUSIONS

We presented a new scalable technique for ME in MPEG encoding. The scalability can be exploited to reduce the computational effort over a large range, making our system feasible for low-cost mobile MPEG systems. The ME technique has been split into a precomputation stage and an approximation stage. Optionally, a refinement stage can be added to come to the quality of a conventional MPEG encoder (or even outperform it). In the precomputation stage, we used a simple recursive block matcher to find rather good motion estimates because the frames are processed in time-consecutive order. In the approximation stage, vector fields are scaled and added or subtracted (thus having multi-temporal references), which is less complex than performing advanced vector searches. The computation of e.g. the backward ME can be omitted to save computational effort and memory bandwidth usage.

Our proposal allows scaling of the computational effort by a factor of 14, resulting in a global variation of 7 dB SNR in picture quality. Furthermore, our system slightly outperforms a 32×32 full-search ME in quality, albeit at a much lower computational effort. Future work will include investigations for obtaining optimal dynamic GOP structures based on the precomputation stage of our new method and explore the effect of scalable DCT [1] and ME in a completely scalable MPEG encoding system.

6. REFERENCES

[1] S. Mietens, P.H.N. de With and C. Hentschel, "New DCT Computation Algorithm for Video Quality Scaling," *IEEE Int. Conf. on Image Proc. (ICIP 2001)*, vol. 3, pp. 462–465, Oct. 2001.

[2] C. Hentschel *et al.*, "Scalable Algorithms for Media Processing," *IEEE Int. Conf. on Image Processing (ICIP 2001)*, vol. 3, pp. 342–345, Oct. 2001.

[3] R. Li, B. Zeng and M.L. Liou, "A new Three-Step Search Algorithm for Block Motion Estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

[4] J.Y. Tham *et al.*, "A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation," *IEEE Trans. on Circuits and Syst. f. Video Techn.*, vol. 8, no. 4, pp. 369–377, Aug. 1998.

[5] P.H.N. de With, "A simple recursive motion estimation technique for compression of HDTV signals," *IEE Int. Conf. Image Proc. and its Applic. (IPA'92)*, pp. 417–420, 1992.

[6] G. de Haan *et al.*, "True-Motion Estimation with 3-D Recursive Search Block Matching," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 3, no. 5, pp. 368–379, Oct. 1993.