ADAPTIVE MOTION ESTIMATION ALGORITHM FOR H.264/AVC

Svetislav Momcilovic, Nuno Roma, Leonel Sousa

INESC-ID / IST TULisbon
Rua Alves Redol 9, 1000-029, Lisboa – PORTUGAL
{Svetislav.Momcilovic,Nuno.Roma,Leonel.Sousa}@inesc-id.pt

ABSTRACT

A new adaptive motion estimation algorithm is proposed in this paper. When compared with other fast search approaches, such as the H.264/AVC oriented EPZS algorithm, this algorithm significantly speeds up the motion estimation procedure and substantially decreases the memory requirements. Moreover, it also makes use of significantly fewer memory accesses, still maintaining its coding quality performances in what concerns both the obtained bit rate and PSNR. As a consequence, the proposed algorithm proves to be specially adequate to be implemented in most embedded systems with restricted computational and power requirements that are often adopted by portable and battery supplied devices.

Index Terms— Video Coding, Motion Estimation, Fast Search Algorithms

1. INTRODUCTION

Motion Estimation (ME) is one of the most important parts of a video encoder. It is used to exploit the temporal redundancy in video sequences, in order to reduce the amount data of the encoded video. However, it is also one of the most computationally intensive blocks. In the H.264/AVC coding standard, it represents up to 80% of the whole set of computations.

Block Matching (BM) is one of the most adopted techniques to perform ME. The H.264/AVC standard uses, for the first time, multiple reference frames for this ME procedure (up to 5 reference frames) [1]. It also includes 7 different sub-block sizes for ME (from 4×4 to 16×16 pixels). Hence, for each 16×16 pixels Macroblock (MB), different ME modes may be considered, according to the adopted Mode Decision (MD) strategy. For both the MD and the ME procedures, it is often used a cost function $\mathrm{J}(x, y)$ based on the Rate-Distortion Optimization (RDO) theory and defined as:

$$\mathrm{J}(x, y) = \mathrm{SAD}(x, y) + \lambda \mathrm{B}(x - x_0, y - y_0), \qquad (1)$$

where (x, y) is the candidate Motion Vector (MV), (x_0, y_0) is the predicted MV for the current MB, $\mathrm{SAD}(x, y)$ represents the matching distortion value calculated as a Sum of Absolute Differences (SAD), $B(x - x_0, y - y_0)$ is the number of bits required to encode the candidate MV and λ is a Lagrange multiplier [2].

Among the several possible ME algorithms that may be adopted, the Full-Search (FS) BM algorithm provides the optimal solution, at the cost of a huge computational load. On the other hand, fast but sub-optimal algorithms, such as the Three-Step-Search (TSS), the Diamond Search (DS) etc., compute the best matching candidate by guiding the search procedure using predefined search patterns. One of such fast algorithms that is often adopted in the H.264 standard is the Hybrid Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) [3]. This algorithm makes use of a complex pattern structure based on four different search patterns.

However, even these fast algorithms still require a significant amount of computations. Often, they can only reach real-time encoding with very fast coding architectures. As a consequence, fast ME algorithms are usually combined with adaptive schemes that make use of prediction and early stopping techniques. The aim of these techniques is to predict the starting point that should be used by the search procedure and to skip unnecessary search points as soon as the matching is good enough, respectively. In the H.264/AVC it is adopted an adapted variant of the Enhanced Predictive Zonal Search (EPZS) algorithm [4]. This algorithm has shown to be very robust, by using a significant number of predictors and a very complex pattern structure. However, its memory and computational requirements are very high, making it difficult to be implemented in many encoder architectures.

A new adaptive ME algorithm special suited to be used with the H.264/AVC standard is proposed in this paper. This algorithm significantly speeds up the ME procedure, when compared with other fast search algorithms. It also substantially decreases the memory requirements of the H.264/AVC oriented EPZS algorithm. When compared with the original EPZS algorithm, it makes use of significantly fewer memory accesses, but still maintains its coding quality performances, in what concerns both the obtained bit rate and PSNR.

The rest of the paper is organized as follows. In section 2 the proposed adaptive ME algorithm is formulated. In this section it is also presented a detailed description of the adopted techniques, namely, prediction techniques (subsection 2.1), adaptive search patterns (subsection 2.2) and early stopping techniques (subsection 2.3). Section 3 presents the experimental results and section 4 concludes the paper.

2. PROPOSED ALGORITHM

The proposed adaptive ME algorithm is the result of the combined effect of three different techniques : *i) prediction techniques*, whose main purpose is the selection of the best start-

```
P  1. Test median predictor
E       if median threshold is satisfied then go to End
P  2. Test all predictors
E       if threshold is satisfied then go to End
A  3. Pattern center = best predictor
A  4. if median threshold is satisfied then pattern = square
       else pattern = adaptive cross,
            radius = max.  spatial pred.  coordinate
A  5. Apply pattern forever
E       if threshold is satisfied then go to End
E       if early stopping is satisfied then go to step 6
A       if best check point == central point then
          if pattern == adaptive cross then
            if radius>2 then radius --
            else pattern = square
          else go to step 6
        else center = best check point
A  6. if median threshold is not satisfied and not last then
        pattern center = second best predictor,
        last = true, go to step 5
P  7. Update adaptive temporal and block-based predictors
   8. End.
```

Fig. 1. Proposed adaptive ME algorithm.

ing search point in order to accelerate the search for the best matching MB; *ii) adaptive search pattern*, in order to further optimize the search procedure; *iii) early-stopping techniques*, whose main aim is to stop the search as soon as the obtained result is good enough and it is not useful to prosecute with it. As it will be shown in the following, the combination of these techniques will provide a substantial reduction of the involved computations and memory accesses, without introducing any significant quality loss in the output video stream.

In fig. 1 it is formulated the proposed adaptive ME algorithm. Each line of the presented pseudo-code was marked with the corresponding technique that is exploited in that stage of the algorithm: spatial and temporal predictions ($\mathcal{P}$), adaptive search pattern ($\mathcal{A}$) and early-stopping techniques ($\mathcal{E}$). A detailed description of the application of all these techniques will be presented in the following subsections.

2.1. Prediction techniques

Prediction techniques are mostly based on the inherent spatial and temporal correlation of the MVs of neighboring or homologous MBs, in order to accelerate the search for the best matching MB.

2.1.1. Area based temporal prediction

Temporal prediction techniques are usually based on the information concerning the MVs from previous frames. Most common approaches use the MVs corresponding to the homologous or surrounding MBs as temporal predictors, but require a considerable amount of storage space. In contrast, the proposed reduced memory approach for temporal prediction is based on the two following assumptions: *(a)* temporal prediction improves the required prediction significantly less than spatial prediction (described in the next subsection) and its importance is only justified by the high complementarity that it provides when combined with spatial prediction; *(b)* in homologous temporal prediction it is assumed that the current MB will keep the same movement of the homologous MB from the previous frame; however, part of the image that

Fig. 2. Frame partition for area based temporal prediction.

belonged to this homologous MB has already moved and consequently, the stored information might only be useful if the same motion pattern involves the surrounding MBs.

Hence, it can be shown that temporal prediction may be significantly improved if the mean value of more neighboring MBs is also considered. Moreover, to make temporal prediction usable by hardware motion estimators, the required memory should be reduced. Based on this assumption, this paper proposes the following new area based temporal prediction approach: *i)* the image is first divided into equal-sized square-shaped areas (see fig. 2); *ii)* for all these areas the mean MVs are calculated and used as temporal predictors. Additionally, if the MB under processing is close to the edge of one of such areas, the mean MVs of the adjacent areas may also be taken into account.

By considering assumption *(b)*, it was also decided to perform the temporal prediction using only the maximum considered block size for the MBs (16×16 pixels). This granulation of the ME blocks presents a inherent trade-off between prediction precision and ME memory requirements.

2.1.2. Spatial prediction within ME sub-blocks

Besides the usual spatial prediction set consisting of the MVs of the left, the upper and the upper-right neighbor MBs, the proposed ME algorithm also considers the zero and the median value of all these predictors.

Moreover, the proposed algorithm also considers a second prediction set when more than one MV is considered for each MB, by performing the ME for different block sizes. Such prediction is carried out based on the assumption that the MVs already estimated for the sub-blocks of a given MB can be very good predictors for the remaining ME procedures that still have to be carried out for the remaining sub-blocks.

However, many of these MVs are frequently null. If all these MVs were to be considered, the amount of required storage space would increase, for each MB, with the number of considered MVs and reference frames. On the other hand, many of the stored MVs would be useless, since they would be either equal to zero or to some other MV of the same MB. As a consequence, the proposed algorithm only considers the non-null MVs with the minimum ME cost. To evaluate this cost, the ME block size is taken into account. As an example, the ME cost of an 8×8 sub-block is multiplied by 4 when it is compared with the cost of a 16×16 pixels sub-block.

2.2. Adaptive search pattern

In contrast to the complex search patterns used by the newest ME approaches, the Adaptive Rood Pattern Search (ARPS) algorithm [5] proposes a simple and still highly adaptive search procedure. It starts with an adaptive radius cross pattern, where the displacement between each check point of the large pattern is multiplied by an adaptive radius parameter. The value of this parameter is obtained by evaluating the greatest coordinate of the best predictor MV. This algorithm will then keep this pattern until the best checking point is found in the center of the pattern, which will make it to switch to the standard 3×3 square pattern.

A modified and more robust variant of this pattern is proposed in this paper. For the adaptive radius it is used the absolute value of the greatest MV coordinate among the spatial predictor MVs. However, if the best matching point is found in the center of the search pattern, the adaptive radius is further decreased until it reaches the value of 2. At this instant, the search pattern switches to the smallest one. Moreover, the adaptive radius pattern is only used if the current minimum ME cost is greater than the threshold value used for the median predictor; otherwise, only the small search pattern is applied.

The described search procedure is presented on Fig. 3. The search pattern starts with an adaptive radius value of 3. The first step is presented with the symbol '◯', while the second step is represented with the symbol '×' and the symbol '□' was used to represent the final step. The intermediate steps corresponding to the cases when the best matching point was found in the center of the search pattern were skipped, for the sake of clearness of the figure. The main advantages of this pattern are mainly concerned with its high adaptivity, as well as its simplicity, which make it specially suited for hardware motion estimators.

Moreover, just like the variant of the EPZS algorithm adopted in H.264/AVC, an additional search pattern applied around the second best predictor is also considered in the proposed algorithm. This pattern additionally improves the search procedure, trying to eliminate possible negative effects of just using one center prediction. However, this additional search is only applied if the current minimum ME cost is greater than the threshold value used for the median predictor.

The main purpose of the combination of these two described strategies (adaptive radius approach and second best predictor search pattern) is to further decrease the possibility of the search procedure to be trapped by a local minimum.

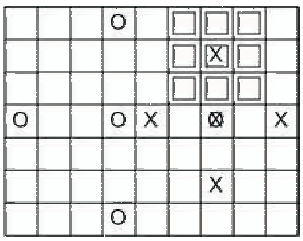

Fig. 3. Proposed adaptive radius search pattern.

2.3. Early stopping methods

The main purpose of the usual threshold methods is to stop the searching procedure as soon as the obtained result is good enough and it is not useful to prosecute with the search.

In the presented algorithm, a new adaptive threshold method is proposed. The considered threshold is based on the normalized value of the minimum cost function that is obtained for the MBs belonging to the same 16×16 pixels area, $B_c(x, y)$. This value is then multiplied by a coefficient γ, depending of the ME state.

$$T_h = \gamma \cdot B_c(x, y) \tag{2}$$

The maximum value of the coefficient γ is used right after the selection of the median predictor, provided that for all neighboring MBs non-null MVs are obtained. The minimum value for this coefficient is used for the smallest block sizes (8×4, 4×8 and 4×4), since the search procedure for these smaller blocks is not so expensive. In all the remaining cases, the value of $\gamma = 1.0$ is adopted.

2.3.1. Rate-distortion based early stopping method

The cost function adopted by the H.264/AVC standard is based on RDO [2]. This ME cost also includes the number of bits required to encode the MV, in order to optimize the trade-off between using greater versus smaller block sizes. In fact, if smaller blocks are adopted, the resulting ME will give rise to more accurate matching, at the cost of using a greater amount of bits to encode more MVs.

Considering this analysis, an additional early stopping method for small sub-blocks ME is proposed. According to this approach, the search procedure should be kept moving towards the candidate with minimum cost as long as the decreasing rate of the considered cost function $\mathrm{J}(x, y)$ is big enough to compensate for the increase of bits required to encode the resulting MVs. Hence, the following inequality is used for the early stopping criterion, whenever $\Delta\mathrm{J}(x, y) > 0$:

$$\frac{\Delta \mathrm{J}(x, y)}{B_c(x, y)} \frac{A(x_0, y_0)}{B(x - x_0, y - y_0)} \cdot \alpha < 1 \tag{3}$$

where α is a calibration coefficient and $B(x - x_0, y - y_0)$ is the number of bits required to encode the candidate MV. The first fraction relates the advantage of decreasing the matching error with the best obtained ME cost in the MBs belonging to the same 16×16 pixels area. $A(x_0, y_0)$ denotes the local activity function, defined as the median value of the MVs corresponding to the left, upper and upper-right neighboring MBs. With this function, it is bridged the significant differences between optimal calibration coefficients that are found when different types of video sequences are considered.

3. EXPERIMENTAL RESULTS

To assess the performance of the proposed ME algorithm, it was integrated within the JVT reference software of an H.264/AVC video encoder JM12.0 [6]. The algorithm was

Table 1. Comparative results in what concerns the coding quality, the bit-rate and the computational load.

	Algorithm	PSNR [dB]	bit rate [kbps]	CP/frame [x10^3]
	FS	38.94	26.01	10070.24
	UMHexagonS	38.89	25.95	76.76
akiyo (QCIF)	EPZS	38.92	26.13	68.58
	Proposed	38.91	25.97	33.29
	FS	36.07	89.20	10070.24
	UMHexagonS	36.06	89.25	175.67
foreman (QCIF)	EPZS	36.07	88.77	108.38
	Proposed	36.09	89.89	84.56
	FS	33.97	297.11	10070.24
	UMHexagonS	33.97	296.65	237.14
mobile (QCIF)	EPZS	33.99	298.02	120.80
	Proposed	33.98	297.74	86.24
	FS	35.89	63.71	10070.24
	UMHexagonS	35.88	63.52	113.71
weather (QCIF)	EPZS	35.87	63.83	91.54
	Proposed	35.89	64.00	59.92
	FS	34.98	287.38	10070.24
	UMHexagonS	34.95	289.27	213.93
stefan (QCIF)	EPZS	34.96	287.52	124.48
	Proposed	34.95	290.96	90.87
	FS	35.19	1076.5	40280.96
	UMHexagonS	35.19	1081.64	874.91
bus (CIF)	EPZS	35.21	1071.55	481.51
	Proposed	35.18	1098.52	342.94

then compared with the EPZS, the UMHexagonS and the Full Search BM approaches. All these algorithms were used to encode the first 100 frames of several QCIF and CIF video sequences at 30 fps and considering 5 reference frames.

In Table 1 it is presented the obtained comparative results in terms of the PSNR (coding quality), the resulting bit-rate (coding efficiency) and the average number of Check Points (CPs) that were performed per frame - CP/frame (computation load). The evaluation of the matching distortion requires different amounts of computations, depending on the ME sub-block size. As a consequence, 16×8 pixels blocks are counted as 0.5 of this cost, 8×8 count 0.25 and so on. For the evaluation of the total amount of checked points, only P encoded frames were considered. From the presented results it can be clearly seen that the proposed algorithm saves up to 50% of the computational load required by the EPZS algorithm and even more when compared with the UMHexagonS, while still providing similar coding quality levels and bit-rates.

One additional key advantage provided by the proposed algorithm is concerned with the low memory requirements. By using the described area-based temporal prediction and the adapted block-size prediction mechanisms, the considerable memory space requirements that characterize other approaches that have been proposed in the literature were eliminated. In Table 2 it is presented the comparative results for these memory requirements for the considered algorithms (w and h denote the number of MBs in each direction and r represents the number of reference frames). The memory required for the MB and search area under processing was not considered.

Table 2. Additional memory space required by the considered algorithms; each MV occupies 2 bytes and the value of the cost function occupies 3 bytes in memory.

Algorithm	FS	UMHexagonal	EPZS	Proposed
Mem[Bytes]	-	-	$2w(h \cdot r + 1)$	$110 + w$

For the EPZS algorithm, all the memory locations required to apply the temporal and spatial predictors were considered, whose exact size depends on the number of reference frames. For the proposed algorithm, it was considered the memory required for the area based temporal prediction using a 5×5 grid, using two MVs and two cost values for each sub-block prediction. The same amount of memory used by the EPZS algorithm was considered for the standard spatial prediction. From Table 2 it can be seen that the proposed algorithm significantly reduces the required memory space, making it specially suited to be implemented by hardware motion estimators. In fact, the usual standard homologous temporal and spatial prediction for the QCIF format occupies almost 2 times more memory than the proposed algorithm.

4. CONCLUSIONS

A new adaptive ME algorithm was proposed. This algorithm significantly speeds up the ME procedure and substantially decreases the memory requirements, when compared with other fast search approaches, such as the H.264/AVC oriented EPZS algorithm, still providing similar coding quality and bit-rate performances. As a consequence, it proved to be specially suited to be implemented in most embedded systems with restricted computational and power requirements that are often adopted by portable and battery supplied devices.

5. REFERENCES

[1] ITU-T, *ITU-T Recommendation H.264, "Advanced Video Coding for Generic Audiovisual Services"*, May 2003.

[2] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[3] "Fast integer pel and fractional pel motion estimation for AVC," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-F016*, Dec. 2002.

[4] H.-Y.C. Tourapis and A.M. Tourapis, "Fast motion estimation within the H.264 codec," in *Proc. of International Conference on Multimedia and Expo, (ICME'03)*, Baltimore, MD, July 2003, vol. 3, pp. 517–520.

[5] Y. Nie and K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Trans. on Image Proc.*, vol. 11, no. 12, pp. 1442–1449, Dec. 2002.

[6] *JVT Reference Software - unofficial version 12.0*, http://iphome.hhi.de/suehring/tml/download.