Accepted in IEEE Int. Conference on Image Processing (ICIP 2015), 2015

AN ADAPTIVE SEARCH ORDERING FOR RATE-CONSTRAINED SUCCESSIVE ELIMINATION ALGORITHMS

Luc Trudeau, Stéphane Coulombe, Christian Desrosiers

Department of Software and IT Engineering
École de technologie supérieure, Université du Québec
Montréal, Québec, Canada

ABSTRACT

This paper proposes a solution for the problem of unnecessary cost function evaluations, found when combining the successive elimination algorithm with a spiral scan search ordering. Our experiments show that the implementation of such a combination inside the HEVC reference software leads to unnecessary cost function evaluations. On the tested video sequences, an average of 3.46% unnecessary cost function evaluations was measured. Considering only small block sizes (e.g., 4×8 and 8×4), this average rises to 8.06%. To solve this problem, we propose an adaptive scan ordering of block matching candidates within the search area. When used with our early termination threshold, the proposed approach will only evaluate necessary cost functions, without impacting rate-distortion.

Index Terms— Successive elimination algorithms, Motion estimation, HEVC.

1. INTRODUCTION

In the context of video compression, motion estimation algorithms improve compression by exploiting spatiotemporal predictability. Given a cost function, they select the best candidate block from a search area in one or many anchor frames to serve as a predictor for the content of the current block. This operation is often referred to as block matching.

Recent demands for beyond-HD video formats (e.g., 4K or 8K), the emergence of multiview video content, and feature-rich video compression standards are all factors that require video encoders to consider more block sizes, more anchor frames, and use bigger search areas [1]. Modern video encoders evaluate a much greater amount of block matching candidates than their former counterparts.

The High Efficiency Video Coding (HEVC) standard [2], also known as H.265, is an example of a feature-rich video encoding standard. H.265/HEVC was designed to provide approximately 50% bit-rate savings for equivalent perceptual quality when compared to the H.264/MPEG-4 AVC standard [3]. However, it also contributes to a significant increase in the computational burden related to motion estimation.

An Exhaustive Search Algorithm (ESA) performs motion estimation by evaluating the cost function for each block matching candidate inside the search area of every anchor frame. The high computational complexity incurred by evaluating the cost of all possible candidates allowed by the HEVC standard limits practical applications of the ESA in modern encoders.

Algorithms designed to reduce this computational complexity are classified according to whether or not they preserve optimality. Those that do not preserve optimality often rely on the assumption of a monotonically increasing match criterion around the location of the optimal candidate block. When this assumption does not hold, the accuracy of the motion estimation algorithm is reduced, as it will converge towards a local minimum. Modern algorithms in this class include zonal search algorithms [4, 5], which first evaluate a set of predictors in order to constrain a local diamond or square search to a very narrow zone of the search area.

Optimality-preserving algorithms employ known inequalities to filter out block matching candidates. These inequalities allow to quickly determine the candidates whose cost functions cannot be smaller than that of the current best candidate. In this situation, evaluating the cost is unnecessary and only wastes computational resources. Recent algorithms in this class append more efficient filtering criteria to the successive elimination algorithm (SEA) proposed in [6]. For example, [7, 8] propose to partition blocks to improve filtering efficiency.

An important fact, not considered in the work of Li and Salari [6], is that not all block matching candidates require the same amount of bits to encode their motion vector. Optimal encoding must consider not only the prediction accuracy of a candidate, but also the number of bits required to encode its vector. In encoding standard recommendations, the trade-off

This work was funded by Vantrix Corporation and by the Natural Sciences and Engineering Research Council of Canada under the Collaborative Research and Development Program (NSERC-CRD 428942-11). Computations were made on the supercomputer Colosse from Université Laval, managed by Calcul Québec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), NanoQuébec, RMGA and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT). Emails: {luc.trudeau, stephane.coulombe, christian.desrosiers}@etsmtl.ca

between prediction accuracy and motion vector cost is defined using the Lagrange multiplier (λ); the Lagrange multiplier is included in the motion estimation cost function, but is not taken into consideration by the original SEA algorithm.

In [9], Coban and Mersereau modified the SEA filtering criterion to take into account the number of bits required to encode the motion vector of a block matching candidate. This modification is in line with the HEVC encoder-side description of the test model reference software [10], where the optimal matching candidate block is the best rate-constrained match.

The SEA filtering criterion can also be improved via the candidate search ordering used during motion estimation. Spiral search ordering outperforms raster search ordering by evaluating better block matching candidates earlier in the search process, which in turn filters out more block matching candidates. Spiral search ordering is considered state-of-the-art, and is used in many implementations of SEA-based algorithms [8, 9, 11]. However, Trudeau et al. [12] showed that a spiral search ordering, in a rate-constrained context, will perform unnecessary cost function evaluations. To avoid this, the motion vector cost of the search ordering must increase monotonically, otherwise the SEA filtering criterion is weakened.

In this paper, we extend the work of [6], [9] to support the H.265/HEVC standard and implement it in the HEVC reference software. However, as described in [12], this leads to unnecessary cost function evaluations. To resolve this problem, we propose an approach to adapt the search ordering of block matching candidates. We also describe a motion estimation algorithm that can adapt the search ordering, and when combined with the proposed early termination threshold, will only perform necessary cost function evaluations. To our knowledge, this is the first time a motion estimation algorithm capable of only performing necessary cost function evaluations is proposed. The result of this motion estimation algorithm is the same as with the ESA.

This paper is organized as follows: Section 2 describes the rate-constrained successive elimination and motivates the need for rate-constrained search orderings. The proposed dynamic approach to generate an adaptive search ordering is presented in section 3. In section 4, the experimental procedure and results are discussed. Finally, a summary of the contributions of this work and results are given in section 5.

2. RATE-CONSTRAINED SUCCESSIVE ELIMINATION ALGORITHMS

The Successive Elimination Algorithms (SEA) use the triangle inequality to filter out block matching candidates whose cost function cannot be lower than the current cost of the best block matching candidate. This leads to computational savings of 85% when compared to ESA[6].

The inequality proposed in [6] is written as:

$$|\sum B - \sum C(\mathbf{x}_i, \mathbf{y}_i)| \leqslant \sum |B - C(\mathbf{x}_i, \mathbf{y}_i)| \,, \quad (1)$$

where B is a matrix of the pixel values inside the current block and $C(\mathbf{x}_i, \mathbf{y}_i)$ is a matrix of the pixel values of the ith candidate block located at position $(\mathbf{x}_i, \mathbf{y}_i)$ in the search area. We will refer to the left hand side of this equation as the absolute difference of sums (ADS).

The right hand side contains the sum of the absolute differences (SAD) between the pixel values of the current block and those of the ith candidate block.

While the complexity of computing $\sum B$ and $\sum C(\mathbf{x}_i, \mathbf{y}_i)$ might appear to be equivalent to that of computing $\sum |B - C(\mathbf{x}_i, \mathbf{y}_i)|$, using the fast calculation of block sums proposed in [6], B and $C(\mathbf{x}_i, \mathbf{y}_i)$ are precomputed and only require table lookups during motion estimation. As explained in [6], the overhead for precaculating these sums is negligible when compared to the savings gained.

In [9], Coban and Mersereau added the rate constraint to the successive elimination algorithms. In a rate-constrained video encoder, this constraint must be added to the SEA in order to produce correct results. Failure to do so could impair the motion estimation module. They also found that the constraint produced a more effective filtering. The rate-constrained inequality is defined as:

$$\begin{aligned} &|\sum B - \sum C(\mathbf{x}_i, \mathbf{y}_i)| + \lambda R(\mathbf{x}_i, \mathbf{y}_i) \\ &\leqslant \sum |B - C(\mathbf{x}^*_{i-1}, \mathbf{y}^*_{i-1})| + \lambda R(\mathbf{x}^*_{i-1}, \mathbf{y}^*_{i-1}) \,, \end{aligned} \quad (2)$$

where λ is the Lagrange multiplier (for the HEVC standard, the recommended function to compute λ is described in [10]). The $R(x, y)$ function, often referred to as the rate, returns the number of bits required to encode the motion vector of the block matching candidate at position (x, y). The term $(\mathbf{x}^*_i, \mathbf{y}^*_i)$ is the current best candidate, considering the candidates from 0 to $i - 1$ in the search ordering, and is such that:

$$\begin{aligned} \forall n \in \{0, \ldots, i\} \Big(&\sum |B - C(\mathbf{x}^*_i, \mathbf{y}^*_i)| + \lambda R(\mathbf{x}^*_i, \mathbf{y}^*_i) \\ &\leqslant \sum |B - C(\mathbf{x}_n, \mathbf{y}_n)| + \lambda R(\mathbf{x}_n, \mathbf{y}_n) \Big) \,. \end{aligned} \quad (3)$$

3. ADAPTIVE SEARCH ORDERING

Implementing the successive elimination algorithm combined with a spiral search ordering in the H.265/HEVC HM reference software can considerably reduce – but not eliminate – unnecessary cost function evaluations. For example, a spiral search ordering applied to a bad predicted motion vector would cause multiple unnecessary cost function evaluations. Even with a good motion vector prediction, the spiral ordering does not follow the monotonically increasing rate rule defined in [12]. This will lead to unnecessary cost function evaluations as the SEA threshold in equation 2 will vary according to the difference in rate between the best block matching candidate and the current candidate.

While exponential Golomb codes are no longer used to encode motion vectors, the method proposed by Trudeau et al. [12] for H.264 can be extended to HEVC. In HEVC, the bit length of the exponential Golomb encoding of the motion vector is recommended as a fast estimate of the motion vector cost for the rate-constrained motion estimation algorithm (as seen in the HEVC reference software [10]).

The new class of search orderings, proposed in [12], also known as rate-constrained search orderings, are based on the monotonically increasing rate rule

$$R(\mathbf{x}_i, \mathbf{y}_i) \geqslant R(\mathbf{x}_{i-1}, \mathbf{y}_{i-1}) \ . \tag{4}$$

This rule states that block matching candidates must be evaluated by increasing order of motion vector bit length, hence the name rate-constrained search ordering. We can see from Figure 1 that multiple search orderings can be derived from this rule, as the ordering of candidates with the same motion vector bit length can be intermixed.

22	22	20	20	18	12	18	20	20	22	22
22	22	20	20	18	12	18	20	20	22	22
20	20	18	18	16	10	16	18	18	20	20
20	20	18	18	16	10	16	18	18	20	20
18	18	16	16	14	8	14	16	16	18	18
12	12	10	10	8	2	8	10	10	12	12
18	18	16	16	14	8	14	16	16	18	18
20	20	18	18	16	10	16	18	18	20	20
20	20	18	18	16	10	16	18	18	20	20
22	22	20	20	18	12	18	20	20	22	22
22	22	20	20	18	12	18	20	20	22	22

Fig. 1: Motion vector bit lengths for the block matching candidates of a subset of the search area. The gray square is the predicted motion vector. Multiple rate-constrained search orderings are possible, as block matching candidates with the same motion vector bit length can be combined in any order.

With multiple search orderings comes the question of which ordering is optimal, which is not trivial to answer. Because the efficiency of the search ordering is greatly influenced by the content of the video sequence.

A search is SEA-optimal if the cost function is evaluated only for the block matching candidates such that

$$\begin{aligned} &|\sum B - \sum C(\mathbf{x}_i, \mathbf{y}_i)| + \lambda R(\mathbf{x}_i, \mathbf{y}_i) \\ &\leqslant \sum |B - C(\hat{\mathbf{x}}, \hat{\mathbf{y}})| + \lambda R(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \ , \end{aligned} \tag{5}$$

where $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is the best candidate over the entire search area. Thus, the search is SEA-optimal if the number of cost function evaluations is equal to the number of block matching candidates, where the rate-constrained ADS (RCADS) is less than the best rate-constrained SAD (RCSAD) found inside the search area (only those candidates have a chance to be optimal). Indeed, because of equation 1, it is unnecessary to evaluate candidates that do not meet equation 5. Furthermore, although we do not know the best candidate needed to apply equation 5, the MotionEstimation function is our proposed adaptive method for an SEA-optimal search.

The parameters of the MotionEstimation function are: **B**, the image to encode; **C**, the anchor frame; (bx, by), the block coordinates; ord, the search ordering; **sB**, the sum buffer of **B**, and **sC**, the sum buffer of the **C**.

By using table lookups (lines 1 and 10), we can determine the RCADS (line 12) of each block matching candidate in the search area. The candidates are then sorted by increasing value of RCADS (line 16). Starting with the lowest RCADS candidate, we successively evaluate the cost function until the later exceeds the lowest cost (lines 19-20), at which point evaluating the cost function of the remaining block matching candidate becomes irrelevant (because of equation 2). This procedure is SEA-optimal because no candidate for which RCADS is higher than the best cost is evaluated (this is obvious from the RCADS-sorted scan order). It is also obvious that the proposed procedure ensures that the optimal value is found.

```
MotionEstimation($\mathbf{B}, \mathbf{C}, bx, by, ord, \mathbf{sB}, \mathbf{sC}$)
 1  $sumB \leftarrow \mathbf{sB}[bx][by]$
 2  $x \leftarrow ord[0].x$
 3  $y \leftarrow ord[0].y$
 4  $bestCost \leftarrow \mathrm{SAD}(\mathbf{B}, \mathbf{C}, bx, by, x, y) + \lambda \times \mathrm{R}(x, y)$
 5  $bestVector \leftarrow ord[0]$
 6  $numCont \leftarrow 0$
 7  for $i \leftarrow 1$ to $ord.length - 1$
 8      $x \leftarrow ord[i].x$
 9      $y \leftarrow ord[i].y$
10      $sumC \leftarrow \mathbf{sC}[bx + x][by + y]$
11      $vCost \leftarrow \lambda \times \mathrm{R}(x, y)$
12      $rcasd \leftarrow |sumB - sumC| + vCost$
13      if $rcasd \leqslant bestCost$
14          $cont[numCont] \leftarrow (rcasd, x, y, vCost)$
15          $numCont \leftarrow numCont + 1$
16  $cont \leftarrow \mathrm{Sort}(cont)$
17  for $i \leftarrow 0$ to $numCont - 1$
18      $rcasd \leftarrow cont[i].rcasd$
19      if $rcasd > bestCost$
20          return $bestVector$
21      else
22          $x \leftarrow cont[i].x$
23          $y \leftarrow cont[i].y$
24          $cost \leftarrow cont[i].vCost$
25          $cost \leftarrow cost + \mathrm{SAD}(\mathbf{B}, \mathbf{C}, bx, by, x, y)$
26          if $cost < bestCost$
27              $bestCost \leftarrow cost$
28              $bestVector \leftarrow (x, y)$
```

We can further reduce the processing complexity by pruning the block matching candidates that will be sorted (lines 13-14). Assuming that we evaluate the cost function of a candidate, by applying its cost in equation 2, we can filter out candidates with higher RCADS. The closer the cost function is to the best cost, the better the filter will be. For convenience, we will use the cost function of the predicted motion vector (line 4) as an upper bound in equation 5 to replace the best cost (which is unknown). In the worst case, the predictor will not be effective, and all candidates will need to be sorted. However, this will not affect the number of cost function evaluations (just the number of candidates to sort, which has a small impact on performance).

4. EXPERIMENTAL RESULTS AND DISCUSSION

To test our hypotheses, we implemented the rate-constrained SEA with a spiral scan search ordering and the proposed approach in the H.265/HEVC HM 13.1 reference software [10]. By comparing the cost function evaluation of both approaches, we could determine the percentage of unnecessary cost function evaluations performed by the RCSEA with a spiral search ordering.

Table 1 presents detailed results of our experiment for the first 100 frames of standard Class C (832×480) video sequences ("*Basketball Drill*", "*Party Scene*", "*BQ Mall*" and "*Race Horses*"). The results are presented by block sizes and by QP values. We used the main profile with the following alterations: 5 reference frames, disabled asymmetric motion partitions, full pixel precision motion estimation and full search motion estimation.

As stated in [12], changing the search ordering has negligible to no impact on rate-distortion as all candidates are considered, only in a different order.

From the results in table 1, we can see that the proposed algorithm is more effective for smaller block sizes. This is due to the fact that smaller blocks comprise fewer pixels, which leads to more precise ADS values. These values filter out more unnecessary cost function evaluations. Since most SEA-based algorithms partition bigger blocks using multiple small partitions to improve filtering efficiency [7, 8, 11, 13], they would benefit significantly from the proposed method.

As the QP increases, the effectiveness of the proposed algorithm also increases. This is analogous to the findings of [9], and is caused by an increase in the value of the Lagrange multiplier. This in turn increases the ratio between the weighted number of bits required to encode the motion vector and the prediction error. When this occurs, the rate constraint becomes more significant and allows more block matching candidates to be filtered.

Table 1 shows that the proposed search ordering is, on average, more efficient with sequences that contain important and unpredictable movement ("*Basketball Drill*" and "*Race Horses*"), than with those with more predictable sequences. Unpredictable sequences lead to less precise motion vector predictions, and for them, hard-coded search orderings, such as spiral scan, will search around a bad starting point leading to unnecessary cost function evaluations. In the same context, by sorting block matching candidates, the proposed adaptive approach exploits the relative precision of the RCADS, allowing candidates around the true motion vector to be considered earlier in the search process.

Table 1: Percentage of unnecessary cost function evaluations made by a rate-constrained SEA with a spiral scan search ordering in the H.265 HM reference software compared to the proposed method

Block Size	QP	Basket	Party	Mall	Horses
4×8, 8×4	22	8.29	5.58	3.35	7.59
8×8	22	4.74	3.56	2.33	6.34
8×16, 16×8	22	3.59	2.60	1.66	5.97
16×16	22	3.22	2.15	1.23	5.80
16×32, 32×16	22	2.94	1.94	0.96	5.44
32×32	22	2.61	1.60	0.77	4.93
32×64, 64×32	22	2.14	1.18	0.60	3.87
64×64	22	1.89	0.60	0.31	2.89
4×8, 8×4	27	10.99	6.73	3.25	8.46
8×8	27	5.94	4.33	2.14	6.49
8×16, 16×8	27	3.56	3.01	1.53	5.79
16×16	27	2.87	2.21	1.16	5.50
16×32, 32×16	27	2.57	1.91	0.93	5.18
32×32	27	2.23	1.58	0.75	4.73
32×64, 64×32	27	1.81	1.16	0.56	3.79
64×64	27	1.62	0.61	0.31	2.88
4×8, 8×4	32	13.12	7.95	3.30	10.10
8×8	32	7.78	4.99	1.97	6.99
8×16, 16×8	32	4.39	3.30	1.38	5.86
16×16	32	2.89	2.29	1.06	5.42
16×32, 32×16	32	2.51	1.83	0.88	5.10
32×32	32	2.14	1.46	0.72	4.62
32×64, 64×32	32	1.76	1.07	0.52	3.80
64×64	32	1.46	0.58	0.27	2.79
4×8, 8×4	37	15.35	9.19	3.31	12.45
8×8	37	9.06	5.51	1.84	7.33
8×16, 16×8	37	5.19	3.43	1.21	5.50
16×16	37	3.07	2.18	0.90	4.93
16×32, 32×16	37	2.30	1.68	0.78	4.66
32×32	37	1.93	1.29	0.66	4.27
32×64, 64×32	37	1.61	0.98	0.50	3.46
64×64	37	1.37	0.49	0.28	2.72

5. CONCLUSION

In this paper, we proposed an algorithm that can dynamically adapt the search ordering of the motion estimation module, and an early termination threshold that guarantees to only perform necessary cost function evaluations. Our experiments show that, without our algorithm, an implementation of the rate-constrained successive elimination algorithm using a spiral scan search ordering in the H.265/HEVC HM reference software would lead to an average of 3.5% unnecessary cost function evaluations. In some instances, the proposed method can reduce the percentage of cost function evaluations up to 15%.

6. REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6316136

[2] ISO/IEC JTC 1/SC 29/WG 11, *High efficiency video coding*, ITU-T H.265 Recommendation, Rev. 1.0, 2013.

[3] ITU-T SG16 Q.6 and ISO/IEC JTC 1/SC 29/WG 11, *Advanced video coding for generic audiovisual services*, ITU-T H.264 Recommendation, Rev. 1.0, 2003.

[4] A. Tourapis, O. Au, and M. Liou, "Predictive motion vector field adaptive search technique (PMVFAST): enhancing block-based motion estimation," *Proc. SPIE Visual Communications and Image Processing*, vol. 4310, pp. 883–892, 2001. [Online]. Available: http://dx.doi.org/10.1117/12.411871

[5] A. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," *Proc. SPIE Visual Communications and Image Processing*, vol. 4671, pp. 1069–1079, 2002. [Online]. Available: http://dx.doi.org/10.1117/12.453031

[6] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Transactions on Image Processing*, vol. 4, no. 1, pp. 105–7, Jan. 1995. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18255420

[7] X. Gao, C. J. Duanmu, and C. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 501–504, Mar. 2000. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18255420

[8] C. Zhu, W.-S. Qi, and W. Ser, "Predictive fine granularity successive elimination for fast optimal block-matching motion estimation," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 213–221, Feb. 2005.

[9] M. Coban and R. Mersereau, "A fast exhaustive search algorithm for rate-constrained motion estimation," *IEEE Transactions on Image Processing*, vol. 7, no. 5, pp. 769–773, May 1998. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18276290

[10] K. McCann, B. Bross, W. Han, and I. Kim, "JCTVC-O1002 High Efficiency Video Coding (HEVC) test model 13 (HM 13) encoder description," *JCT-VC, Tech. Rep*, no. November, 2013.

[11] M. Yang, H. Cui, and K. Tang, "Efficient tree structured motion estimation using successive elimination," *Vision, Image and Signal Processing, IEE Proceedings -*, vol. 151, no. 5, pp. 369–377, Oct 2004.

[12] L. Trudeau, S. Coulombe, and C. Desrosiers, "Rate distortion-based motion estimation search ordering for rate-constrained successive elimination algorithms," in *2014 IEEE International Conference on Image Processing (ICIP 2014)*, Paris, France, Oct. 2014, pp. 3175–3179.

[13] T. Toivonen and J. Heikkila, "Fast full search block motion estimation for H.264/AVC with multilevel successive elimination algorithm," in *2004 IEEE International Conference on Image Processing (ICIP 2004)*, Singapore, Oct. 2004, pp. 1485–1488.