SUB-PARTITION REUSE FOR FAST OPTIMAL MOTION ESTIMATION IN HEVC SUCCESSIVE ELIMINATION ALGORITHMS

Luc Trudeau, Stéphane Coulombe, Christian Desrosiers

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

ABSTRACT

In the context of motion estimation (ME) for video coding, the rate-constrained successive elimination algorithm (RCSEA) safely eliminates candidate motion vectors while preserving the optimal candidate chosen by the block matching algorithm (BMA). This paper describes a technique for reusing ME information from rectangular to square prediction units in order to reduce the search area without altering the optimal candidate chosen by the BMA. Our experiments show that, on average, when this optimization is combined with the RCSEA in the HEVC HM encoder reference software, the number of sum of the absolute differences (SAD) operations drops by 94.9%, resulting in a speedup of 6.13x in full search mode. Although identical coding decisions cannot be guaranteed when multiple optimal solutions exist, the average impact on BD-PSNR is 0.0002 dB.

***Index Terms*—** Successive elimination algorithms, Block matching algorithm, Motion estimation, HEVC.

1. INTRODUCTION

To improve the compression gains provided by predictive coding, video coding standards, such as H.264/MPEG-4 advanced video coding (AVC) [1] and H.265/high efficiency video coding (HEVC) [2], have considerably increased the domain of the motion estimation (ME) function. By domain, we refer to the range of values that can be used when performing motion compensation (MC). Consequently, these standards allow the use of MC for more block sizes and over more reference frames. In addition, the recommended size of the search area used by ME algorithms has also increased [3]. The magnitude of this domain, combined with the computational complexity of evaluating candidates, forces modern encoders to use suboptimal algorithms, such as the popular zonal approaches [4, 5].

In [6], Li and Salari proposed the successive elimination algorithm (SEA), an algorithm which considerably reduces the burden of ME. First, it computes the absolute difference of sums (ADS) over the entire domain. Then, by exploiting the triangle inequality between the ADS and the sum of the absolute differences (SAD), it eliminates candidates by transitivity, thus avoiding costly SAD operations where possible. In contrast, we refer to the subset of filtered candidates as the codomain of the ME function. Performance-wise, the appeal of the ADS is due to the fact that its sums can be stored and reused between blocks and search areas.

Coban and Mersereau proposed the rate-constrained successive elimination algorithm (RCSEA) in [7]. Implementations of this algorithm have been presented for H.264 [8, 9]. To our knowledge, except for our previous works presented in [10] and [11], no work has been published so far on RCSEA for HEVC.

As shown in our previous work [10], if the candidates are ordered by rate, the rate constraint can shrink the search area, thus reducing the codomain to parts of the search area that satisfy the rate constraint. In [11], we showed that the smallest codomain is obtained by sorting the candidates by ADS value. In this work, we reduce the codomain of square prediction units (PUs) even more, by reusing information from previous ME operations performed over rectangular PUs.

The original contributions of this work are as follows:

- We present a new technique for reusing motion estimation information from rectangular PUs inside square PUs (section 3).
- We integrate this new information into the early termination mechanism of RCSEA that we proposed in [10] (section 4).

To facilitate the understanding of our contributions, we present an overview of SEA and RCSEA in section 2. We experimentally confirm that this new optimization significantly reduces the number of SAD operations performed while preserving the optimal candidate chosen by the block matching algorithm (BMA) (section 5).

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 Guillimin from McGill University, 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

2. SUCCESSIVE ELIMINATION ALGORITHM

In this section, we give a brief overview of the SEA's transitive elimination phase in a rate-constrained context.

Let $s \in \{\mathbb{S}, \mathbb{V}, \mathbb{H}\}$ be the partitioning shape of a PU, which can either be a square ($\mathbb{S}$), a vertical rectangle ($\mathbb{V}$) or a horizontal rectangle ($\mathbb{H}$), as shown in Fig. 1. A square PU contains one partition referred to as 0, whereas rectangular PUs have two partitions referred to as 0 and 1.

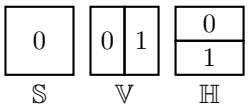

Fig. 1: The partitioning shapes of a PU, a square ($\mathbb{S}$), a vertical rectangle ($\mathbb{V}$) and a horizontal rectangle ($\mathbb{H}$). The first partition is 0 and if a second partition exists, it is indexed as 1.

Let $B_{s,p}$ be the block of shape s at partition p in a PU. $B_{s,p}$ is made up of $M_s \times N_s$ pixels. These pixels are accessed as $B_{s,p}(m,n)$. Let $C_{s,p}$ be the search area related to $B_{s,p}$. Let $C_{s,p,x,y}$ define the candidate block corresponding to the motion vector differential (MVD) (x,y) measured against the motion vector prediction (MVP) of $B_{s,p}$. Let $\mathcal{S}_{s,p}$ be the set of all (x,y) in the search area of $C_{s,p}$ which is centered at the position defined by the MVP of $B_{s,p}$.

The SAD evaluates the error between $B_{s,p}$ and a candidate $C_{s,p,x,y}$ as follows:

$$\mathrm{SAD}(s,p,x,y) = \sum_{m=0}^{M_s-1}\sum_{n=0}^{N_s-1} \left|B_{s,p}(m,n) - C_{s,p,x,y}(m,n)\right| . \tag{1}$$

When performing block matching (BM), the cost function being minimized is defined as follows:

$$\mathrm{J}(s,p,x,y) = \mathrm{SAD}(s,p,x,y) + \lambda R(x,y) , \tag{2}$$

where λ is the recommended HEVC Lagrange multiplier [3], and $R(x,y)$ returns the number of bits required to encode the (x,y) motion vector (MV). Another recommendation of [3], is to use signed exponential Golomb codes to quickly estimate the number of bits required to encode a MV during BMA.

The contribution of the SEA to BM is that it filters out candidates that cannot produce better results than the current best [6, 7]. This is achieved by exploiting the triangle inequality between the SAD and the ADS:

$$\begin{aligned}\left|\sum_{m=0}^{M_s-1}\sum_{n=0}^{N_s-1} B_{s,p}(m,n) - \sum_{m=0}^{M_s-1}\sum_{n=0}^{N_s-1} C_{s,p,x,y}(m,n)\right| \\ \leqslant \sum_{m=0}^{M_s-1}\sum_{n=0}^{N_s-1} \left|B_{s,p}(m,n) - C_{s,p,x,y}(m,n)\right| .\end{aligned} \tag{3}$$

This inequality can be used when successively evaluating MV candidates to perform transitive elimination, as in:

$$\begin{aligned}\mathrm{SAD}(s,p,\hat{x},\hat{y}) + \lambda R(\hat{x},\hat{y}) &\leqslant \mathrm{ADS}(s,p,x,y) + \lambda R(x,y) \\ &\leqslant \mathrm{SAD}(s,p,x,y) + \lambda R(x,y) ,\end{aligned} \tag{4}$$

where $(\hat{x},\hat{y})$ is the position of the current best candidate.

In other words, for a given candidate, at position (x,y), if the rate-constrained ADS value of that candidate is superior to the rate-constrained SAD of the current best candidate, then it is impossible for the rate-constrained SAD value of that candidate to be smaller than that of the current best candidate. Given the rate-constrained SAD value of one candidate in the search area, all candidates with a rate-constrained ADS value greater than the given rate-constrained SAD can safely be eliminated without altering the optimal candidate chosen by the BMA. This outlines the RCSEA as proposed in [7].

3. INFORMATION REUSE BETWEEN PU SHAPES

Conventional approaches evaluate PU partitioning shapes in the order $\mathbb{S} \rightarrow \mathbb{V} \rightarrow \mathbb{H}$, as described in the mode decision section of [3]. This ordering is well-suited for suboptimal algorithms, since decisions related to skipping partitioning shapes can be taken early on. However, in the context of an optimal algorithm, this ordering offers no advantage. More favorable orderings are $\mathbb{V} \rightarrow \mathbb{H} \rightarrow \mathbb{S}$ and $\mathbb{H} \rightarrow \mathbb{V} \rightarrow \mathbb{S}$ as they allow the reuse of computed values from rectangular partitions in the square partition.

Information reuse between partitioning shapes can be performed as follows:

$$\mathrm{SAD}(\mathbb{S},0,x,y) = \mathrm{SAD}(\mathbb{V},0,x,y) + \mathrm{SAD}(\mathbb{V},1,x,y) . \tag{5}$$

This also applies for $\mathbb{H}$; the previously computed SADs of both rectangular partitions at a given position can be summed up to obtain the SAD of the square PU at that position. This is valid under a *constant search area* assumption: $\mathcal{S}_{\mathbb{V},0} = \mathcal{S}_{\mathbb{V},1} = \mathcal{S}_{\mathbb{S},0}$ and $\mathcal{S}_{\mathbb{H},0} = \mathcal{S}_{\mathbb{H},1} = \mathcal{S}_{\mathbb{S},0}$.

The main limiting factor of this approach is the SEA. As previously explained, the SEA filters out many SAD operations, which are therefore not available for reuse. Although it is possible to manage missing SADs, the high percentage of SAD operations filtered out by the SEA, combined with the management overhead, makes such an approach impractical.

One useful piece of information that is unaffected by the SEA is the minimum SAD denoted (SAD^*), defined as:

$$\mathrm{SAD}^*(s,p) = \min_{(x,y)\in\mathcal{S}_{s,p}} \mathrm{SAD}(s,p,x,y) . \tag{6}$$

This information is useful because summing up the SAD^*s of the partitions of $\mathbb{V}$ or $\mathbb{H}$ yields a lower bound for the SAD^* of $\mathbb{S}$:

$$\mathrm{SAD}^*(\mathbb{S},0) \geqslant \mathrm{SAD}^*(\mathbb{V},0) + \mathrm{SAD}^*(\mathbb{V},1) . \tag{7}$$

$$\mathrm{SAD}^*(\mathbb{S},0) \geqslant \mathrm{SAD}^*(\mathbb{H},0) + \mathrm{SAD}^*(\mathbb{H},1) . \tag{8}$$

These inequalities are valid under the *constant search area* assumption.

Let SAD^Ω be the lower bound of the $\mathrm{SAD}^*(\mathbb{S},0)$, such that:

$$\begin{aligned}\mathrm{SAD}^\Omega = \max\big(&\mathrm{SAD}^*(\mathbb{V},0) + \mathrm{SAD}^*(\mathbb{V},1), \\ &\mathrm{SAD}^*(\mathbb{H},0) + \mathrm{SAD}^*(\mathbb{H},1)\big).\end{aligned} \tag{9}$$

By using the highest lower bound, we have a tighter lower bound to the $\text{SAD}^*(\mathbb{S}, 0)$. Per the definition of SAD^Ω and using Eq. (7) and Eq. (8):

$$\text{SAD}^\Omega \leqslant \text{SAD}^*(\mathbb{S}, 0) \leqslant \text{SAD}(\mathbb{S}, 0, x, y),\ \forall\ (x, y) \in \mathcal{S}_{\mathbb{S},0}\ . \quad (10)$$

4. IMPROVED EARLY TERMINATION FOR $\mathbb{S}$

The increasing rate rule, presented in [10], states that BM candidates shall be ordered by increasing rate ($R(x, y)$). This ordering allows for early termination of the RCSEA. It follows from Eq. (4), and since $\text{SAD}(s, p, x, y) \geqslant 0$, that early termination can occur when

$$R(x, y) \geqslant \frac{\text{SAD}(s, p, \hat{x}, \hat{y})}{\lambda} + R(\hat{x}, \hat{y})\ . \quad (11)$$

When Eq. (11) holds, the candidate cannot be an optimal solution as its weighted rate ($\lambda R(x, y)$) exceeds the current minimum rate-constrained SAD. According to the increasing rate rule, the algorithm can terminate, as all subsequent candidates cannot be optimal solutions.

The $\text{SAD}(s, p, x, y)$ found in Eq. (4) is not in Eq. (11), since no assumption can be made about the SAD of subsequent candidates beyond the fact that they are greater than or equal to zero. A lower bound greater than zero would permit early termination. This is indeed what we propose now for $\mathbb{S}$. In Eq. (10), we demonstrated that $\text{SAD}(\mathbb{S}, 0, x, y) \geqslant \text{SAD}^\Omega$. Therefore, for $\mathbb{S}$, termination occurs when

$$R(x, y) \geqslant \frac{\text{SAD}(\mathbb{S}, 0, \hat{x}, \hat{y}) - \text{SAD}^\Omega}{\lambda} + R(\hat{x}, \hat{y})\ . \quad (12)$$

Per Eq. (10), when Eq. (12) holds, the candidate cannot be an optimal solution, as its weighted rate ($\lambda R(x, y)$) added to the lower bound of the $\text{SAD}(\mathbb{S}, 0, x, y)$ exceeds the current minimum rate-constrained SAD. According to the increasing rate rule, the algorithm can terminate; all subsequent candidates cannot be optimal solutions, as they cannot have a SAD lower than SAD^Ω. The higher the value of SAD^Ω, the sooner the termination in Eq. (12) occurs, compared to Eq. (11).

This improvement does not alter the optimal candidate chosen by the BMA. As shown in the next section, this constraint on the size of the search area considerably reduces the codomain of BM over $\mathbb{S}$.

5. EXPERIMENTAL RESULTS

The test conditions and software configurations used in our experiments conform to the common test conditions and software reference configurations of the JCT-VC [12]. The encoder software runs the main profile with 8-bit coding and Low Delay P settings.

The only changes to the standard configuration files are that enable full search, and the fast encoder decision (FEN) and asymmetric motion partitions (AMP) are disabled. We disabled the latter only to simplify the implementation of the proposed methods, but AMP and RCSEA are compatible. All tests were performed on the first 100 frames of the sequences specified by Bossen [12], for classes: B (1920×1080), C (832×480), and D (416×240).

5.1. Comparison with HEVC HM Full Search

In the first part of Table 1, we compare the SAD savings, the encoding time speedup and the Bjøntegaard delta peak signal-to-noise ratio (BD-PSNR) [13] of the unmodified HM reference encoder, version 16.6, against the proposed solution also implemented in version 16.6 of the HM reference encoder. The values shown are averaged from the results for quantization parameters (QPs): 22, 27, 32 and 37. The speedup is measured as the ratio between the encoding time of the unmodified HM reference software (T_{HM}) and the encoding time of the HM reference software with the proposed solution (T_{Proposed}):

$$\text{SpeedUp} = \frac{T_{\text{HM}}}{T_{\text{Proposed}}}\ . \quad (13)$$

The SAD savings are measured as the relative difference between the number of SAD operations of the HM reference encoder ($\#\text{SAD}_{\text{HM}}$) and the number of SAD operations of the proposed solution ($\#\text{SAD}_{\text{Proposed}}$):

$$\text{SAD Savings} = \frac{\#\text{SAD}_{\text{HM}} - \#\text{SAD}_{\text{Proposed}}}{\#\text{SAD}_{\text{HM}}}\ . \quad (14)$$

The proposed solution, implemented with the RCSEA in the HM reference encoder, eliminates 94% of SAD operations on average, and is approximately 6 times faster than the original HM encoder. We observed that the bit streams produced by the proposed solution and the unmodified HM software encoder are not identical. This is due to two factors: the ordering differences between same-cost candidates during ME and the ordering differences of same-cost partitioning shapes during the coding of the PU. In other words, when multiple global minimums exist, either when choosing an MV or when choosing a partitioning shape, both encoders might not pick the same one. That being said, as shown in Table 1, the difference in BD-PSNR is negligible (less than 0.004 dB).

5.2. Comparison with RCSEA

The second part of Table 1 shows the encoding time speedup, the total SAD operation savings, and the SAD operation savings for square PUs for the contributions of this paper alone. We compare the proposed solution to our previous work [10] adapted to HEVC, and implemented in version 16.6 of the HM reference encoder. We did not compare it to [11], as speedups would be biased, because of the time required to sort candidates by ascending ADS.

Compared to our previous work adapted to HEVC, the proposed solution eliminates, on average, 19.8% more SAD operations, which is directly attributable to the 63.7% savings of square SAD operations, resulting in a speedup of approximately 1.23. From Fig. 1, we see that a PU requires 5 ME

		Prop. vs HM			Prop. vs [10] (for HEVC)		
Class	Sequence name	Speedup	SAD Savings	BD-PSNR	Speedup	SAD Savings	$\mathbb{S}$ SAD Savings
B (1920×1080)	Kimono	6.30	96.7%	0.0006	1.15	14.9%	45.6%
	ParkScene	6.42	95.8%	0.0014	1.35	25.7%	79.4%
	Cactus	7.07	96.3%	0.0018	1.27	21.8%	67.9%
	BQTerrace	5.92	94.6%	-0.0020	1.36	26.3%	81.9%
	BasketballDrive	6.05	95.4%	0.0016	1.23	20.2%	64.0%
C (832×480)	RaceHorses C	4.73	92.7%	0.0011	1.13	14.8%	50.2%
	BQMall	6.70	95.5%	-0.0008	1.18	16.0%	53.1%
	PartyScene	4.68	91.6%	-0.0003	1.27	19.9%	66.2%
	BasketballDrill	5.59	95.4%	-0.0026	1.24	19.3%	61.0%
D (416×240)	RaceHorses	4.56	93.0%	-0.0030	1.15	12.9%	43.1%
	BQSquare	8.75	96.1%	0.0032	1.34	27.6%	90.4%
	BlowingBubbles	6.78	95.2%	-0.0020	1.22	20.7%	68.1%
	BasketballPass	6.18	95.4%	-0.0011	1.20	17.8%	56.9%
	Overall	6.13	94.9%	0.0002	1.23	19.8%	63.7%

Table 1: Comparison of the proposed solution with the HEVC HM reference encoder software (Prop. vs. HM; see section 5.1). Comparison of the proposed solution with our previous work adapted to HEVC (Prop. vs. [10] (for HEVC); see section 5.2).

searches, one of which is square. It could be assumed, that square SAD operations account for one fifth of the total count of SAD operations. Thus, it could be expected that the SAD savings (column 7 of Table 1) would represent at most one fifth of the total square SAD savings (column 8). That is however not the case, due to the fact that because of the early termination mechanism, the square SAD operation savings make up about one-third of the total count of SAD operations savings.

To elaborate, square blocks are twice the size of their rectangular counterparts. As a result, square SADs are, more or less, twice as big as rectangular SADs. From Eq. (11), it therefore follows that early termination requires twice the rate. Because of exponential Golomb codes, doubling the rate exponentially increases the size of the search area. However, based on an assumption of spatial-temporal correlation, we can assume that doubling the rate also exponentially increases the efficiency of rate-constrained transitive elimination. From this, we can expect the number of SAD operations to double. Thus, the weighted ratio of square SAD operations is about one third of rectangular SADs (a good approximation of the savings observed in Table 1).

Fig. 2 shows that the square SAD operation savings increase when the QP increases. This is in line with the findings of Coban and Mersereau [7] to the effect that an increase in the Lagrange multiplier has a direct impact on transitive elimination in a rate-constrained context. As demonstrated, this property still holds for the proposed solution.

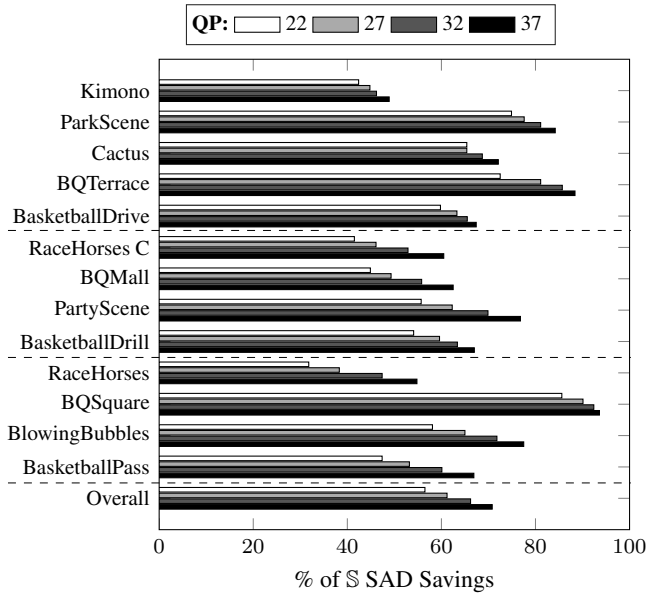

Fig. 2: Percentage of square SAD operation savings, per sequence, for the proposed solution, when compared to our previous work [10] adapted to HEVC

6. CONCLUSION

In this paper, we describe a technique for reusing motion estimation information from rectangular to square PUs. Our experiments show that on average, when compared to the HEVC HM reference encoder software, the proposed solution reduces the number of SAD operations by 94.9%, resulting in a 6.13x speedup. For square partitions, this exceeds our previous work adapted to HEVC by an average of 63.7%, resulting in a 1.23x speedup.

This work is not meant to be compared with suboptimal ME approaches. Our contribution resides in a reduction of the codomain of the ME function, while preserving the optimal candidate. Our work counteracts the increased domain of the ME function imposed by modern video encoding standards. We hope that this work can serve as the foundation for novel approaches to both optimal and suboptimal ME.

7. REFERENCES

[1] ITU-T SG16 Q.6 and ISO/IEC JTC 1/SC 29/WG 11, "ITU-T recommendation H.264: Advanced video coding for generic audiovisual services," 2003.

[2] ISO/IEC JTC 1/SC 29/WG 11, "High efficiency video coding," 2013.

[3] Ken McCann, Chris Rosewarne, Benjamin Bross, Matteo Naccari, Karl Sharman, and Gary J Sullivan, "High efficiency video coding (HEVC) test model 16 (HM 16) improved encoder description," Tech. Rep. October, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Strasbourg, France, 2014.

[4] A.M. Tourapis, O.C. Au, and M.L. 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.

[5] A.M. 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.

[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.

[7] M.Z. Coban and R.M. 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.

[8] 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.

[9] 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.

[10] Luc Trudeau, Stéphane Coulombe, and Christian 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.

[11] Luc Trudeau, Stéphane Coulombe, and Christian Desrosiers, "An adaptive search ordering for rate-constrained successive elimination algorithms," in *2015 IEEE International Conference on Image Processing (ICIP 2015)*, Québec, Canada.

[12] Frank Bossen, "Common test conditions and software reference configurations Output," *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG1*, vol. JCTVC-L110, no. 12th Meeting: Geneva, pp. 1–10, 2013.

[13] Gisle Bjøntegaard, "Calculation of average PSNR differences between RD-curves," 2001.