Motion Estimation with Multiple Matching Criteria

Gabriel Lemes S. L. de Oliveira
Department of Electrical Engineering
University of Brasília
Brasília, Brazil
Email: gabriel.lemes@gmail.com

Eduardo Peixoto
Department of Electrical Engineering
University of Brasília
Brasília, Brazil
Email: eduardopeixoto@ieee.com

Ricardo L. de Queiroz
Department of Electrical Engineering
University of Brasília
Brasília, Brazil
Email: queiroz@ieee.com

***Abstract*—Block-based motion estimation is the method of choice in most video codecs to exploit temporal redundancy for compression. Since true rate-distortion evaluation for every candidate block is usually impractical, simple estimates are used instead as a matching criterion, e.g., the Sum of Absolute Differences (SAD) between the target and the candidate blocks weighted by its respective motion vector cost in bits. We show that different matching criteria may differ not only in the quality of the resulting motion estimation but may actually offer diverse motion estimates that can be locally selected to better suit the local data characteristics. As a proof of concept, we propose the Double Matching Criteria Algorithm (DMCA), a two-pass algorithm which performs two independent motion estimations for each block with different matching criteria and then locally selects the better one in a true rate-distortion sense. The algorithm is easily parallelizable and out-of-the-box compliant to any coding standard with support for block-based motion compensation. We also propose the Total Absolute Deviation from the Mean (TADM) as a matching criterion to be used along with the SAD in the DMCA framework. Unlike the SAD, which measures the size of the residue in a sense, the TADM is a measure of its dispersion. For demonstration purposes, we implemented the DMCA with the SAD and the TADM as matching criteria in a modified HM reference software encoder for the HEVC standard. We observed significant BD-rate gains with a fully compliant HEVC stream.**

I. Introduction

Most video codecs implement some form of *block-based motion compensation* (MC) to exploit interframe redundancy for compression purposes [1], [2], [3]. The frame to be encoded is divided in a number of blocks and, in order to reduce the necessary number of bits to encode a given target block at a given target quality, the encoder searches among a number of candidate blocks in previously encoded frames for a good match. This stage is called *motion estimation* (ME). The encoder then signals the relative displacement between the target and its matching block, the *motion vector* (MV) for the current target block. With the MV, the decoder can then replicate each prediction block. Most of the time, however, this prediction is not enough to attain the desired target quality, so the block difference between the target and prediction blocks, the *residual*, is also encoded into the bitstream, usually after some form of quantization if lossy compression is allowed. Even though the residual has the same size of the target block itself and requires the extra MV information to be of any value, the rationale is that it is usually more amenable to compression, enough to compensate for the extra MV bits.

In spite of its name and associated heuristics, MC is not concerned with an accurate description of motion. The aim is actually a compact description of the data given a target overall quality, so the minimization of a *rate-distortion* (RD) metrics lends itself as an ideal matching criterion for ME. True RD optimization in the ME stage, however, would require every candidate block to be tentatively used for actual residue quantization and encoding. That would make it extremely expensive in computational resources terms or even impractical when the ME target block might not align with the quantization and encoding blocks, which can be the case for some prediction modes both in the popular H.264/AVC [4] and in the current H.265/HEVC [5] coding standards. Therefore, encoders rely on cheaper estimates, the most popular being a Lagrangian sum of the MV cost in bits with the *Sum of the Absoluter Differences* (SAD) [2], [3]. The SAD is simply the ℓ^1 norm of the residual, so its minimization amounts to making the residual small in some sense. The underlying reasoning is that a small residue would incur in a small distortion even if heavily quantized, while also being heavily concentrated around zero if the target and prediction blocks are similar enough, thus requiring fewer bits for encoding. The MV bits are explicitly considered to avoid large MV's that would probably offset the gains of a small SAD[1].

Other popular matching criteria include the *sum of squared differences* (SSD) and the *Sum of Absolute Hadamard Transformed Differences* (SAHTD) [3], [6]. Alternatives have also been proposed, e.g., the *Normalized Cross-Correlation* [7], the *Pel Difference Classification* [8], and the *Minimized Maximum Error* [9]. Each of these strive for a better matching, all under the same token that a better match will imply a more efficient coding, therefore making that matching criteria inherently superior. In this paper, we argue that even though that may be consistently true on the long run, it may still not be the case for every target block. Different matching criteria might differ not only in the marginal quality of the estimation in an underlying unified sense, they may actually offer radically

[1]We will henceforth omit this MV rate-like regularization term of the cost function for ease of discussion, but it is implied throughout the text.

distinct models for the data which can each be useful for local features. We also show that it happens often enough to merit consideration of multiple matching criteria for ME.

As proof of concept, we propose the *Double Matching Criteria Algorithm* (DMCA), which performs two independent motion estimations for each target block and locally selects the best result in a true RD sense for encoding. We also propose the *Total Absolute Deviation from the Mean* (TADM) as a matching criterion to be used together with the SAD in the DMCA framework. Unlike the SAD which entails a measure of size, the TADM measures the dispersion of the residue, so that its minimization favours dispersion blocks with values more heavily clustered around a central value. It should be noted that the algorithm itself does not interfere with the overall MC framework, affecting only the ME stage. Therefore, it can be made compliant with any coding standard with support for MC. For demonstration purposes, the algorithm was implemented in a modified version of the HM reference software [10] encoder for the HEVC standard [11]. Extensive testing of the DMCA with joint consideration of the SAD and of the TADM under the Common Test Conditions [12] reveals significant and consistent BD-rate [13] gains with a fully compliant HEVC bitstream, even though the TADM itself often underperforms the SAD when both are considered alone.

II. Dispersion Measures for Motion Estimation

As noted, the SAD is simply the ℓ^1 norm of the residue. Also, the SSD, another popular cost function for ME, is the squared ℓ^2 norm, so that its minimization, like the SAD's, also encodes the same heuristics of searching for a small residue. There are, however, other useful matching criteria for motion estimation which have nothing to do with size. In this section, we explore one such successful matching criterion, introduced in the past by rather indirect means.

Blasi *et al* proposed the *Enhanced Inter-Prediction* (EIP) as yet another refinement for the MC framework [14]. In the EIP approach, each candidate block P actually gives rise to a whole family of candidate blocks $P_c(\mathbf{x}) = \Theta(P|\mathbf{x})$ by virtue of an invertible parametric transform $\Theta(\cdot|\mathbf{x})$ with n-dimensional vector parameter $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Rather than directly considering the candidate P, $P' = \Theta(P|\hat{\mathbf{x}})$ was considered instead, where $\hat{\mathbf{x}}$ optimizes P_c with respect to some cost function such as the SAD. For the final matching candidate, $\hat{\mathbf{x}}$ must be coded into the the bitstream along with its respective MV in order to allow the decoder to reproduce the resulting prediction. Also, the transformed residual $R' = P' - T$ is coded instead of the original residual $R = P - T$, with T being the target block.

In practice, the form of the Θ transform is limited by considerations about the bit cost of encoding the vector parameter $\hat{\mathbf{x}}$ as well as the computational cost of the optimization over $\mathbf{x}$ for every candidate. Observe that, through the underlying cost, e.g., the SAD, $\hat{\mathbf{x}}$ might also depend on the target block T. To make the EIP practical, Blasi *et al* then proposed the *Shifting Transformation* (ST) [14], a simple transformation $\Theta(P|s) = P + s$ with a single scalar parameter s, the *shift*. The sum is understood in the sense that s is independently summed to every value in the block P.

A simple closed formula was given for the optimal parameter $\hat{s}$ in terms of the *original*, i.e., non-transformed residual $R = P - T$ when the underlying cost considered was the SSD. In this case, $\hat{s} = -\bar{R}$, the negative of the *mean* value in the residue block R. For the SAD case, an efficient iterative algorithm was given. It was later shown [15] that, in this case too, the optimal value was also a function of the original residue alone: $\hat{s} = -\tilde{R}$, the negative of the *median* value in R. Observe that, in the SSD case, the EIP with ST effectively changes the cost of each candidate block from the simple SSD to

$$SSD_{shift} = \sum_{i=1}^{n} R_i'^2 = \sum_{i=1}^{n} (R_i - \bar{R})^2, \qquad (1)$$

where R_i is the i-th residue value and n is the number of pixels in the target block. In the SAD case, the cost is effectively changed to

$$SAD_{shift} = \sum_{i=1}^{n} |R_i'| = \sum_{i=1}^{n} |R_i - \tilde{R}|. \qquad (2)$$

Furthermore, instead of the original residue R, either $R' = R - \bar{R}$ or $R' = R - \tilde{R}$ is encoded into the bitstream together with its respective shift parameter, either $\hat{s} = \bar{R}$ or $\hat{s} = \tilde{R}$ in the SSD or the SAD case, respectively. Blasi *et al* implemented this approach into the H.264/AVC standard framework and significant gains were observed.

In terms of P' and R', Eq.(1) and Eq.(2) are just the unmodified SSD and SAD, respectively, so that the reasoning behind its minimization is still that of a small residue overall. In terms of P and R, however, Eq.(1) and Eq.(2) have nothing to do with size whatsoever. In fact, its minimization can lead to very large non-transformed residuals. Observe that Eq.(1) is proportional to the sample *variance* of R, while Eq.(2) is proportional to its *mean absolute deviation from the median*, both measures of *dispersion*. That is, aside from the special coding provisions for R' and $\hat{s}$, the EIP with the ST simply changes the distortion term in the ME cost function in order to prioritize residuals that are more densely clustered around a central value instead of a small residual.

Aiming for a low dispersion residue makes some intuitive sense. A lower dispersion might be indicative of a lower entropy, which can lead to more efficient coding. Furthermore, downplaying the DC term of the residue during ME can lead to a better match for the AC terms of the target, which are more heavily quantized in modern codecs such as the H.264/AVC and the HEVC, thus leading to better preservation of textures and edges. The EIP with ST shows this to be indeed the case. However, as we will see in Section V, simply plugging a dispersion cost into the ME stage of an otherwise established codec will not leverage these advantages. In the simple case of the EIP with ST, R carries the exact same information as R' and $\hat{s}$ together. In fact, the very first thing the decoder does in the EIP with ST framework is recovering the exact same R

from R' and $\hat{s}$. Why then there is no gain in simply using Eqs. (1) or (2) for ME and then coding R directly, eschewing the extra bits necessary to code $\hat{s}$? Because R' is better tuned to the built-in entropy coding schemes of most codecs than R which, as noted, can be very large. The part of R that is not suited for the built-in entropy coding, namely $\hat{s}$, is conveniently dealt with in a separate entropy encoder specialized to its statistics. In effect, the EIP with ST simply changes the matching criteria for ME and tunes the entropy encoder to better match the new residue statistics that emerges. Unfortunately, this later part renders it non-compliant to well established coding standards.

In Section III, we propose a framework in which it is possible to leverage these ME heuristics in a standard compliant fashion. For now, let us encode these heuristics in a more computationally friendly dispersion measure. We propose the *Total Absolute Deviation from the Mean* (TADM) as a dispersion measure for ME:

$$TADM(P,T) = \sum_{i=1}^{n} |R_i - \bar{R}|, \qquad (3)$$

in which $R = P - T$ is the residue block for a given candidate-target pair (P,T) and $\bar{R}$ is its mean value. It has the advantage of avoiding the computations of squares as in Eq.(1) or the computation of the median as in Eq.(2), both of which can be very expensive.

III. Double Matching Criteria Algorithm

In Section II we observed that different matching criteria for ME can be more than simply more refined or more robust versions of each other. They can actually encode profoundly different matching heuristics, which in turn can lead to profoundly different models for the target data. We also observed in Section I that the ideal matching criterion in the form of true RD optimization over every ME candidate is highly impractical. It is however very well suited for deciding over a few alternative models. This is precisely the essence of RD optimization [1].

In this spirit, we propose the *Double Matching Criteria Algorithm* (DMCA), a simple two-pass algorithm which employs RD optimization over two independent motion estimations with different matching criteria. For concreteness, we consider the SAD and the TADM in turns for each pass. Assume a frame to be encoded is divided in non-overlapping *motion estimation blocks* (MEB). An MEB could be, e.g., a CTU or a CU in the HEVC framework [5] or a macroblock in the H.264/AVC framework [4]. Each MEB is first predicted as usual with the SAD cost function. Its *motion parameter set* (MPS) is stored in MPS_{SAD}, which includes the residuals, MV's, mode flags, and whatever else might be relevant for its encoding. That is, what constitutes an MPS might differ depending on what is considered an MEB, e.g., a CTU or a CU in HEVC. Another motion estimation is then carried out again for the same MEB, with the TADM cost function for ME instead of the SAD this time around. Its MPS is stored in MPS_{TADM}. The RD cost J is then assessed for both MPS's, where J should be a true RD cost function or a reasonable approximation thereof, considering the actual distortion after quantization and the bit-rate for every relevant parameter, including MV's, residuals, and associated flags. The best MPS in terms of minimal J is then stored in MPS_{MEB} as the actual MPS to be considered for the current MEB. The algorithm is summarized in Algo. (1).

```
Algo. (1): DMCA
FOR each MEB
    MPS_SAD ← MEB prediction using the SAD cost
    MPS_TADM ← MEB prediction using the TADM cost
    IF J(MPS_TADM) < J(MPS_SAD)
        MPS_MEB ← MPS_TADM
    ELSE
        MPS_MEB ← MPS_SAD
```

Observe that the decoder cannot know and need not know whichever of the parameters MPS_{SAD} or MPS_{TADM} was chosen by the encoder as the actual MPS for each MEB. The encoding process in any case is rigorously the same. Therefore, compliance is guaranteed to any and all coding standards with support for MC as no modifications to the decoder are necessary.

IV. HEVC Implementation

For testing purposes, we implemented the DMCA in a modified version of the HM reference software v16.5 [10] for the HEVC standard [11]. Algo. (1) was specialized so that each CU was considered an MEB for the DMCA purposes. Observe that this can and often will result in a mixed CTU being encoded with both SAD-predicted CU's and TADM-predicted CU's. As noted in Section III, this need not be signalled to the decoder. The generated bitstreams are rigorously compliant to the H.265/HEVC standard.

The SAD prediction round was unmodified from the regular HM encoding. Under common test conditions, this implies the regular SAD cost for integer-pixel precision with TZ search for fast ME, and SAHTD cost for sub-pixel refinement. For the TADM round, the TZ search algorithm was correspondingly adapted to also allow for fast ME, but the sub-pixel refinement was conducted without the Hadamard transform, keeping the TADM cost also for sub-pixel ME.

V. Results

We tested our modified HEVC encoder in a number of sequences recommended in the *Common Test Conditions* (CTC) [12], namely, the sequences listed under classes A, B, C, D, and E. Configurations for all encoding runs was set to the RA-main configurations also recommended under CTC, except for the caveats noted in Section IV.

Performance was assessed in terms of BD-rate [13] gains using the unmodified HM encoder as anchor under the same configurations. Table I summarizes results for a regular ME with TADM cost alone, while Table II summarizes results for the DMCA.

As anticipated in Section II, simply plugging the TADM cost for ME in an otherwise unmodified encoder is not enough

TABLE I

BD-RATE SINGLE PASS TADM AGAINST CONVENTIONAL HEVC.

Class/ Resolution	Sequence	FPS	BD-rate	BD-rate High	BD-rate Low
A 2560x1600	Traffic	30	0.7	1.2	0.1
	PeopleOnStreet	30	0.7	1.1	0.2
	Nebuta	60	0.1	0.1	0.2
	SteamLocomotive	60	0.7	0.1	1.1
B 1920x1080	Kimono	24	-0.1	-0.1	-0.1
	ParkScene	24	0.7	1.0	0.5
	Cactus	50	-0.5	-0.5	-0.6
	BQTerrace	60	0.0	0.0	0.1
	BasketballDrive	50	-0.5	-0.5	-0.6
C 832x480	RaceHorses	30	0.9	0.7	1.0
	BQMall	60	0.0	0.0	0.0
	PartyScene	50	0.0	0.0	0.0
	BasketballDrill	50	-0.2	-0.1	-0.3
D 416x240	RaceHorses	30	2.4	2.5	2.1
	BQSquare	60	0.8	0.5	1.0
	BlowingBubbles	50	1.1	1.1	1.0
	BasketballPass	50	1.2	1.4	0.7
E 1280x720	FourPeople	60	0.3	0.9	0.0
	Johnny	60	0.7	1.4	0.2
	KristenAndSara	60	0.3	0.9	0.0
Mean			0.5	0.6	0.3

TABLE II

BD-RATE OF DMCA AGAINST CONVENTIONAL HEVC.

Class/ Resolution	Sequence	FPS	BD-rate	BD-rate High	BD-rate Low
A 2560x1600	Traffic	30	-0.8	-0.8	-0.7
	PeopleOnStreet	30	-1.2	-1.0	-1.4
	Nebuta	60	-0.6	-0.4	-1.3
	SteamLocomotive	60	-1.1	-0.7	-1.4
B 1920x1080	Kimono	24	-0.8	-0.9	-0.8
	ParkScene	24	-0.6	-0.6	-0.6
	Cactus	50	-1.2	-1.1	-1.1
	BQTerrace	60	-0.7	-0.8	-0.6
	BasketballDrive	50	-1.6	-1.5	-1.7
C 832x480	RaceHorses	30	-1.1	-1.1	-1.2
	BQMall	60	-1.1	-1.0	-1.1
	PartyScene	50	-0.9	-0.9	-0.8
	BasketballDrill	50	-1.1	-1.0	-1.2
D 416x240	RaceHorses	30	-1.2	-1.3	-1.2
	BQSquare	60	-0.7	-0.8	-0.5
	BlowingBubbles	50	-1.0	-1.0	-0.9
	BasketballPass	50	-1.1	-1.0	-1.2
E 1280x720	FourPeople	60	-0.8	-0.9	-0.7
	Johnny	60	-0.9	-1.0	-0.8
	KristenAndSara	60	-1.0	-1.1	-0.9
Mean			-1.0	-0.9	-1.0

to observe any gains. In fact, as we can see in Table I, while it shows some gains in a few sequences, such as Cactus and BQTerrace, it actually shows losses on a regular basis. A simple direct check on the comparative performances on Table I could lead to the dismissal of the TADM as unfit for an ME cost function.

Table II, on the other hand, reveals that joint consideration of the SAD and of the TADM cost functions in the DMCA framework leads to significant gains over the regular SAD encoding for all sequences tested, even though the TADM by itself underperforms the SAD on the long run. It is consistent with our argument that different matching criteria can better adapt to local features of the data.

An earlier preliminary version of the present work [15] showed similar performance gains also on the H.264/AVC framework [16], also with standard compliant encoding. Extensive testing there in a number of configuration scenarios also showed that these gains do not compete with other coding tools. For instance, roughly the same BD-rate gains were observed with varying numbers of reference frames, with and without multiple QP testing, and even with full search ME algorithm, which lends further support for our argument that the coding gains observed with the DMCA do indeed emerge from the motion hypothesis diversity brought in by multiple matching criteria.

VI. Conclusion

Different matching criteria can be more than simply more accurate or more robust alternatives to each other. Even though one can be consistently more reliable than the other on the long run, we showed that they can actually provide fundamentally different prediction models for the data. It can be worth then to adaptively select the matching criteria to better match local features of the data, instead of committing to a single supposedly better criterion.

The advantages of using multiple matching criteria for motion estimation was demonstrated with the introduction of a simple two-pass algorithm, the DMCA. For each target prediction unit, it carries out two independent motion estimations using different matching criteria in each turn, and then selects the more efficient one in a rate-distortion sense for encoding into the bitstream. For concreteness, the TADM was also introduced to be used in turns with the popular SAD in the DMCA framework. Unlike the SAD, which encodes a notion of size for the residual, the TADM is a measure of dispersion. Incidentally, we also showed the usefulness of minimal dispersion of the residue as a matching criterion. Although regular encoding with the TADM alone apparently suggests the SAD to be a superior cost function for motion estimation, joint consideration of both in the DMCA framework led to significant gains in coding performance.

Although the algorithm itself was proposed as a proof of concept, the DMCA is also useful in its own right. It can be easily extended to consider any number of matching criteria and it is easily parallelizable since there is no need for exchange of information between concurrent motion estimation processes. This multi-pass approach is also common in video coding, driving other successful techniques. For example, it is in the same spirit of multiple QP testing, available in the reference software for both the H.264/AVC and for the HEVC coding standards. This can be useful in asymmetric scenarios wherein a video file is encoded once and transmitted or decoded many times. Furthermore, it is out-of-the-box compatible with any standard with support for block-based motion compensation. For instance, significant gains in coding performance were observed in both the H.264/AVC and in the HEVC codecs with strictly compliant bitstreams.

Further research include investigation of other cost functions both old and new in the context of motion estimation with multiple matching criteria. This new approach opens a new research direction for matching criteria since their usefulness can now be evaluated beyond a simple direct comparison of overall coding efficiency. Also of great interest are algorithmic developments to more efficiently leverage the benefits of multiple matching criteria, preferably avoiding the multi-pass approach of the DMCA. Finally, we emphasised compliance to well established standards in this work. However, while introducing the TADM, we noted that different matching criteria can also lead to significantly different statistics of the residue. It is also worth investigating then if fine-tuning of the entropy encoder for each matching criteria, with the appropriate signalling also written to the bitstream, can lead to even greater gains in the DMCA framework.

References

[1] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann, 2012.

[2] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, Wiley, 2010.

[3] M. Wien, *High Efficiency Video Coding: Coding Tools and Specifications*, Springer, 2015.

[4] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, *Overview of the H.264/AVC video coding standard*, IEEE Transactions on Circuits and Systems for Video Technology, v. 13, n. 7, pp. 560-576, 2003.

[5] 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, v. 22, n. 12, pp. 1649-1668, 2012.

[6] K. P. Lim, G. Sullivan, and T. Wiegand, *Text description of joint model reference encoding methods and decoding concealment methods*, JVT-o079, Tech. Rep., April 2005.

[7] S. Kappagantula and K. R. Rao, *Motion compensated predictive coding*, in SPIE 27th Proc. 432, pp. G4-70, 1983.

[8] H. Gharavi and M. Mills, *Block-matching motion estimation algorithm—New results*, IEEE Trans. Circuits Sysr.. vol. 37, no. 5, pp. 649-651, May 1990.

[9] M. J. Chen, L. G. Chen, T. D. Chiueh, Y. P. Lee, *A new block matching criterion for motion estimation and its implementation*, IEEE Trans. Circuits Syst. Video Technol., vol. 5, pp. 231-236, June 1995.

[10] JCT-VC, *Subversion Repository for the HEVC TestModel*, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/

[11] ITU-T, *ITU-T recommendation H.265, High Efficiency Video Coding*, ITU-T, 2014.

[12] F. Bossen, *Common test conditions and software reference configurations*, JCT-VC, Tech. Rep., October 2012.

[13] Gisle Bjontegaard, *Improvements of the BD-PSNR model*, ITU-T SG16/Q6, 35th VCEG Meeting, Doc.VCEG-AI11, 2008.

[14] S. G. Blasi, E. Peixoto and E. Izquierdo, *Enhanced Inter-Prediction via Shifting Transformation in the H.264/AVC*, IEEE Transactions on Circuits and Systems for Video Technology, v. 23, n. 4, pp. 735–740, 2012.

[15] G. L. S. L. de Oliveira, *Motion Compensation With Minimal Residue Dispersion Matching Criteria*, M.S. Thesis, FT, UnB, Brasília, BR, 2016.

[16] ITU-T, *ITU-T recommendation H.264, Advanced Video Coding for Generic Audiovisual Services*. ITU-T, 2014.