POWER EFFICIENT MOTION ESTIMATION USING MULTIPLE IMPRECISE METRIC COMPUTATIONS

In Suk Chong and Antonio Ortega

Signal and Image Processing Institute, University of Southern California, Los Angeles CA 90089

ABSTRACT

In this paper, we propose power efficient motion estimation (ME) using multiple imprecise sum absolute difference (SAD) metric computations. We extend recent work in [18] by providing analytical solutions based on modelling of computation errors due to voltage over scaling (VOS) and sub-sampling (SS). Results show that our solutions provide significantly better performance in the sense of rate increase for fixed QP, e.g., less than 5% increase, while in [18] the rate increase could be as high as 20%. Our analysis also allows us to compare different ME algorithms (e.g., full search vs. a fast algorithm) and SAD computation architectures (parallel vs. serial) in terms of their robustness to imprecise metric computations and their power efficiency. Finally, we demonstrate that additional power savings can be achieved by removing redundancy between the various computations.

Index Terms— voltage over scaling (VOS), error tolerance (ET), matching metric computation (MMC), imprecise computation

1. INTRODUCTION

Multimedia applications represent a major workload on a large number of hand-held devices such as cellular phones and laptops [11, 8], for which power (or energy) is the most important design constraint. Video encoders (e.g., H.264/AVC and MPEG-4) are the most power consuming part of multimedia applications. Within a typical video encoder, we focus on the power efficiency of motion estimation (ME) as it consumes large portion of resources (e.g., 66%-94% in an MPEG-4 encoder [12]).

Algorithmic approaches for power efficient ME [10, 9, 17] are have been studied for a number of years. Recently, a new technique, voltage over scaling (VOS) within ME, has been shown to lead to significant additional power savings [6, 18]: given an existing algorithm, the input voltage (Vdd) for the SAD computation module is set below critical voltage (Vdd_{crit}). A reduction in Vdd by a factor W can lead to power dissipation that is nearly a factor of W^2 lower. A major difference with respect to existing algorithmic methods is that the lower power consumption comes at the cost of *input-dependent soft errors*; lower input voltage increases circuit delay, and the number of basic operations possible for one clock period decreases, thus generating error. In [6], we have shown that these errors due to VOS are either i) concealed by the encoding process (e.g., a motion vector selected to minimize SAD can be correct, even if the SAD itself is incorrect) or ii) can lead to "acceptable" degradation (e.g., a small distortion or rate increase). This acceptable degradation characteristic can be seen as a specific instance of error tolerance (ET), which has been considered in more general settings, including hard hardware faults [1]. Our previous work has demonstrated that image/video compression systems exhibit ET characteristics, *even if no explicit error control block is added* in the presence of VOS [2] (additional work dealing with hard errors due to deterministic faults shows similar results [7, 5]). For example, our simulations demonstrated 37% power savings in the ME process with negligible performance penalty in typical video encoders. As part of our work we developed an analytical model for VOS errors in the ME context.

Recent work [18] has also proposed using VOS to achieve power savings in ME. Error concealment is introduced in this system, by computing additional SAD values using a sub-sampled (SS) version of the original macro-block data, and then using SAD data computed by these two computation modules (VOS and SS) in order to estimate the "true" SAD value for each candidate (a simple threshold method is proposed in [18] to combine information provided by the two modules). The basic idea is then to use two imprecise SAD computations (with different characteristics) instead of one. If the error concealment module (based on SS) consumes much lower power than the VOS module then overall power savings can be achieved, as compared to a technique without error concealment (e.g., [6]). Note that algorithmic methods to approximate the SAD metric with lower computation cost are well known (e.g., see analysis of SS in [13] and references therein), but techniques for SAD computation cost reduction based on VOS are very recent.

In this paper, we study a two-module system such as that proposed in [18]. Our main contribution is to use analytical error models for both VOS [6] and SS [13]. These models lead to novel techniques for combining the SAD values obtained by the two modules, which we show outperform the threshold methods proposed in [18]. In particular we show that the additional error tolerance enabled by error concealment with improved SAD estimation leads to an increase in the range of operating values for Vdd and m, which directly translates into increased power savings. Removing redundancy between computations performed by the two models can further reduce overall power. Furthermore, we use these models to evaluate error tolerance and power efficiency of various ME algorithms (full search, FS, and enhanced predictive zonal search, EPZS [3]) and SAD computation architectures (parallel and serial).

Note that some of our techniques may also apply to environments where multiple different *perfect* metric computations are performed in a noisy environment (e.g., deep submicron noise [16]). With appropriate models, the techniques we discuss may lead to increased resilience (with lower are overhead), as compared to traditional techniques such as triple modular redundancy which uses three the same perfect modules and simple majority voting [14].

We first briefly explain the ME process, and propose a problem formulation where two imprecise metric computations (i.e., due to VOS and SS) are used within the ME process (Section 2). In Section 3, we introduce analytical error models for each computation

This paper is based upon work supported in part by the National Science Foundation under Grant No. 0428940. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

module and propose novel techniques to estimate SAD based on the output of these modules. In Section 4, we provide simulation results to evaluate the solutions. Results show that our new estimators substantially outperform previous work in terms of rate increase for fixed QP; our solutions shows less than 5% increases (when previously proposed techniques led to increases of around 20%). Furthermore, we compare ME algorithms (FS/EPZS) and SAD computation architectures (parallel/serial). We also show that additional power savings can be achieved by removing redundancy between computation modules.

2. MOTION ESTIMATION WITH MULTIPLE IMPRECISE COMPUTATIONS

The ME process comprises a search strategy (ME algorithm) and a matching metric computation (MMC). The search strategy identifies a set of candidate motion vectors (MVs) and then proceeds to compute the matching metric for the candidates and to select the one that minimizes the matching metric (typically SAD). There are several types of hardware architectures to compute the matching metrics, with different levels of parallelism [15]. We will refer to them as MMC architectures. Among those, we mainly use a serial architecture which has M^2 serially connected adders for SAD computation between two $M \times M$ macro-blocks. A parallel architecture is also used for comparison.

For each $M \times M$ macroblock in the current frame, the i-th candidate MV SAD is denoted SAD_i, and we assume there are N candidates. We define I as the candidate index corresponding to lowest SAD [1] ($I = argmin_i(SAD_i)$), so that the minimum SAD is $SAD_{min} = SAD_I$. Here we consider two imprecise SAD computations with VOS and SS (as in [18]). Note that this formulation can be generalized to multiple SAD computations that are different in the sense of being subject to different types of errors.

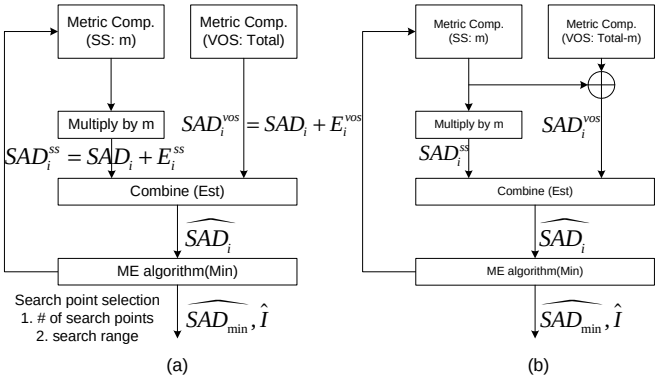

Fig. 1. ME with two imprecise metric computations, Left: Redundant, Right: Non-Redundant

Denote $SAD_i^{ss} = SAD^i + E_i^{ss}$ and $SAD_i^{vos} = SAD^i + E_i^{vos}$, the SADs corresponding to the i-th candidate computed by the SS and VOS modules, respectively (See Figure 1 (a)). These two sets of SAD values are used to estimate the best MV, corresponding to index $\hat{I}$. If $I \neq \hat{I}$, the residual block's distortion (as measured by the SAD) increases by $E_{SAD} = SAD_{\hat{I}} - SAD_I$. This increase in distortion (E_{SAD}) may lead to a rate increase (ΔR) for a given QP. For example, a quadratic model [4] suggests that ΔR is a linear function of E_{SAD}, so that the relative rate increase can be approximated as $\frac{\Delta R}{R} = \frac{E_{SAD}}{SAD_I}$.

Now our goal is to provide a method to find $\hat{I}$ such that E_{SAD} is minimized. We will use the error models for E_i^{ss}, E_i^{vos} (proposed in [6] and [13], respectively). We propose a method that operates in two steps: i) for each candidate find an SAD estimate $\widehat{SAD}_i$ based on information provided by each module

$$\widehat{SAD}_i = f(SAD_i^{ss}, SAD_i^{vos}), \quad (1)$$

and ii) find $\hat{I}$ which minimizes $\widehat{SAD}_i$.

3. PROPOSED SAD ESTIMATORS

In [18] a threshold-based estimator (Est_{Th}) is proposed. If the difference between SAD_i^{vos} and SAD_i^{ss} is larger than Th, then SAD_i^{ss} is chosen as an estimate of SAD_i. Otherwise SAD_i^{vos} is used as an estimate. The same Th is applied to all SAD_i:

$$Th = max_i |SAD_i^{ss} - SAD_i| \quad (2)$$

Note that the threshold is defined in terms of SAD_i, which is not known beforehand, so that an approximate procedure (or training) may have to be developed to estimate (details are not provided in [18]). This approach is a heuristic based on two simple observations about VOS and SS errors; i) the magnitude of VOS errors tends to be large and always leads to SAD values that are *below* the correct ones, and ii) SS errors are usually relatively small compared to VOS errors especially for low sub-sampling factors m, i.e., in cases when a greater percentage of pixel data is used to compute SAD^{ss}. This approach works well for Vdd and m values where errors are relatively small (e.g., relatively large Vdd and $m \leq 4$). As will be shown next, by using models for both SS and VOS errors, it is possible to achieve good performance in a larger range of Vdd and m values, thus further increasing power savings.

3.1. Error Characteristics for VOS and SS

We describe briefly the error models for VOS and SS that have been proposed in [6] and [13], respectively. The VOS error (E_i^{vos}) is a non-positive discrete random variable with values that are multiple of -2^{R_S} for a given SAD_i. Here R_S is the number of basic operations (e.g., full adder operation in ripple carry adder) possible for one clock cycle, which is a non-decreasing function of Vdd: higher Vdd implies large R_S and thus more operations can be completed per cycle, leading to a lower probability of computation error. As a result, VOS error depends on both Vdd and the input characteristics. In summary, we have that $SAD_i^{vos} \leq SAD_i$ and $SAD_i^{vos} = SAD_i$ for $SAD_i < 2^{R_S}$. Refer to [6] for additional details. Note also that the error in computing SAD_i^{vos} is upper bounded by SAD_i; also, from our simulations we observe that $Pr(SAD_i^{vos} \neq SAD_i)$ for small SAD_i is more than 10 times smaller than the average $Pr(SAD_i^{vos} \neq SAD_i)$.

The SS error (E_i^{ss}) can be modelled [13] as a continuous laplacian random variable with parameter λ for given SAD_i, where λ is a function of the sub-sampling parameter m; larger m's correspond to larger λ parameters. We observe that λ varies as a function of SAD_i; thus, for smaller SAD_i an accurate λ may be up to an order of magnitude smaller than the average λ that would be selected for all SADs. Note that for given SAD_i, SS and VOS errors are practically uncorrelated (in our simulation, correlation < 0.02): in what

[1] Note that this could be replaced by another selection metric, e.g., one involving a Lagrangian cost.

follows we assume that SS errors are independent of VOS errors for given SAD_i.

3.2. Adaptive Threshold Estimator and MAP Estimator

An adaptive threshold strategy can be defined based on the observation that in the SS model λ decreases with SAD_i. We can divide the range of SAD values into intervals $r = 1, 2, ..., K$, and associate a threshold Th_r (and corresponding λ_r) to each interval, based on the observed SAD_i^{ss}. Thus, thresholds can be smaller for smaller SAD_i^{ss}.

Additionally, within each interval, we can use a maximum a posteriori (MAP) estimator for SAD_i based on SAD_i^{ss}, SAD_i^{vos}, which can be defined as follows:

$$\begin{aligned}\widehat{SAD}_i &= argmax_x Pr(SAD_i = x | SAD_i^{ss}, SAD_i^{vos}) \\ &= argmax_x Pr(SAD_i^{ss}, SAD_i^{vos} | SAD_i = x) Pr(SAD_i = x)\end{aligned} \tag{3}$$

Assuming that SAD_i is uniformly distributed would lead to a maximum likelihood (ML) estimator. Note that applying this estimator for each candidate index i does not require significant computational complexity. First, $Pr(SAD_i^{ss}, SAD_i^{vos} | SAD_i = x) = Pr(SAD_i^{ss} | SAD_i = x) \cdot Pr(SAD_i^{vos} | SAD_i = x)$, under our assumption of independence of SS and VOS errors for given SAD_i. Second, the above term only needs to be evaluated for a small number of values because the VOS error is a discrete random variable with sparse support. Third, we can further approximate the distribution of VOS errors as follows. Given the probability $p_0^r = Pr(SAD^{vos} = SAD \text{ in } r-th \text{ interval})$ that no VOS errors when SAD^{ss} belongs to the r-th interval, we can approximate all L nonzero errors (multiples of -2^{R_S}) as having the same probability $\frac{1-p_0^r}{L}$. This has negligible effect on the MAP estimator performance, according to our simulations.

In our observation this MAP estimator can occasionally be somewhat sensitive to modelling errors. Thus a more robust estimator would combine the MAP technique with the adaptive threshold estimator. Note also that both the adaptive threshold estimator and the MAP estimator are designed to find the best estimate of SAD_i, and thus are not optimized in terms of our final objective, i.e., minimizing E_{SAD}. We next propose an estimator to address this objective.

3.3. MAX Estimator

Consider a cost function (J) that takes into account the expected value of E_{SAD}:

$$\begin{aligned}J &= \textstyle\sum_k (SAD_k - SAD_{min}) Pr(k = \hat{I}) \\ &= \textstyle\sum_k (SAD_k - SAD_{min}) Pr(\forall i \neq k, \widehat{SAD}_i > \widehat{SAD}_k)\end{aligned} \tag{4}$$

We propose a heuristic estimator to avoid minimizing J directly. Divide the candidates into two sets: $\mathbf{B} = \{k | SAD_k >> SAD_{min}\}$ and $\bar{\mathbf{B}} = \{k | SAD_k \approx SAD_{min}\}$. Then, for $k \in \mathbf{B}$, it is desirable to have $\widehat{SAD}_k > min_s(\widehat{SAD}_s, s \in \bar{\mathbf{B}})$ with high probability. This condition suggests that a reasonable estimator should have a positive bias (to avoid introducing errors in identifying the minimum SAD candidate) and that the bias should increase with the SAD values, i.e., ideally smaller bias for $k \in \bar{\mathbf{B}}$ and larger bias for $k \in \mathbf{B}$. Note that $\widehat{SAD}_k \geq SAD_k^{vos}$, since VOS errors are always negative, but that we do not have a specific upper bound for $\widehat{SAD}_k$ and thus the bias could be arbitrarily large. As a heuristic, we propose to select as an estimator $max(SAD_k^{ss}, SAD_k^{vos})$, which has the desired property of tending to introduce a bias that increases with SAD_k. Note that this is a non-parametric estimator, which means that overall estimation complexity is modest.

4. SIMULATION RESULTS AND DISCUSSION

We now evaluate the performance of our proposed estimators: i) adaptive threshold estimator, ii) MAP combined with adaptive threshold estimator, and iii) MAX estimator. For our experiments we use the FOREMAN sequence with a series of Vdd, QP, and m using an H.264/AVC baseline profile encoder with FS/EPZS ME algorithms and serial/parallel MMC architectures. Only 16×16 block partitions and a single reference were considered for ME. A constant QP was used and rate distortion optimization was turned on. We assign 15 frames to each group of pictures (GOP), and use an IPPP GOP structure. We collect distortion increase ($E_{SAD} = SAD_{\hat{I}} - SAD_I$) data by encoding each GOP with/without errors for different Vdd ($R_S = 10, 12, 14$ where $R_S = 16$ for error free operation), $QP = 10, 20, 30$, and $m = 2, 4, 8$ (note that $m = 4$ was the maximum sub-sampling rate that could be supported with the proposed threshold estimator in [18]). We evaluate the relative rate increase using $\frac{E_{SAD}}{SAD_I}$.

Clearly, in selecting parameters (i.e., $\lambda_r, Th_r, p_0^r, \lambda, Th$) for the various estimators we do not have access to the original SAD. One possible approach would be to use a few blocks per frame for training, and then use the same estimator parameters for all remaining blocks in the frame. However, we observed that estimator parameters can vary significantly within a frame. Thus, as an approximation, we use SAD_i^{vos} as an estimate of SAD_i, and use this to select estimator parameters for each macroblock.

With these settings, we compare the performance of the threshold estimator of [18] and the three new estimators. In the sense of minimizing rate increase, MAX estimator shows best performance, followed by, in order of decreasing performance, MAP combined with adaptive threshold, adaptive threshold, and threshold estimator; performance differences are clearer when FS is used. In Figure. 2, MAX estimator outperforms threshold estimator even for $m = 8$ and $R_S = 10$ where the threshold estimator is known to be suboptimal; MAX estimator shows less than 5% rate increase, while the threshold estimator shows around 20% rate increase. In the other range of parameters all new estimators show reasonable performance; with all estimators leading to less than 5% rate increase. Note that a 20% rate increase would correspond to around $0.5dB$ PSNR loss if the rate were to be kept fixed rate, while 5% rate increase leads to less than $0.1dB$ loss [2]. Also in [18], three step search (TSS) algorithm was used, which is shown to be less resilient to soft errors than EPZS, thus rate increase can be worse if we use TSS instead of EPZS [2]. In the extreme case, when we use FS algorithm, MAX estimator shows similar rate increase, but threshold estimator shows more than 90% increase, which corresponds to more than $2dB$ loss.

We also compare MMC architectures and ME algorithms. Since a EPZS search strategy uses a good prediction algorithm to select a small number of MV candidates, which are already near the minimum SAD point, EPZS has smaller number searching points and search range than the FS algorithm. Thus EPZS shows more resilience to errors due to imprecise computations than the FS case, lower Vdd can be used for EPZS, resulting in better power efficiency. But here we do not consider the inherent difference in complexity, regularity, and memory usage between EPZS and FS algorithm, which is not easy to quantify; FS algorithm has more searching points but has more regular structure and memory usage than EPZS. In case of MMC architecture, parallel architecture shows better performance than serial one because its intermediate nodes has

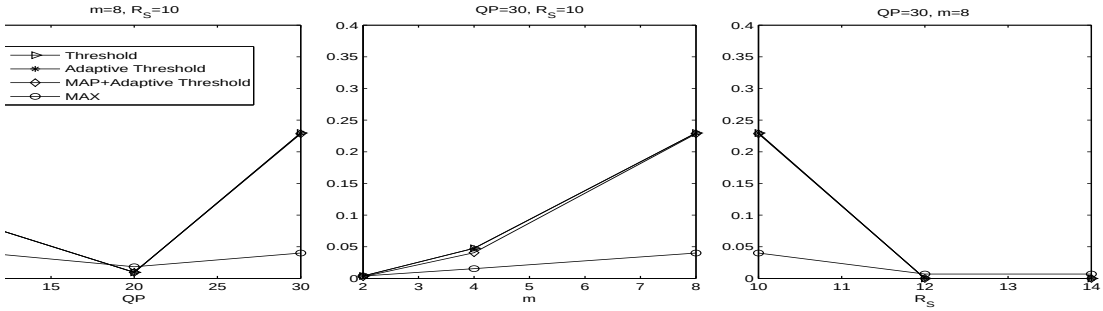

Fig. 2. Comparison of four estimators with EPZS algorithm for various parameters; $m = 2, 4, 8$, $R_S = 10, 12, 14$, and $QP = 10, 20, 30$.

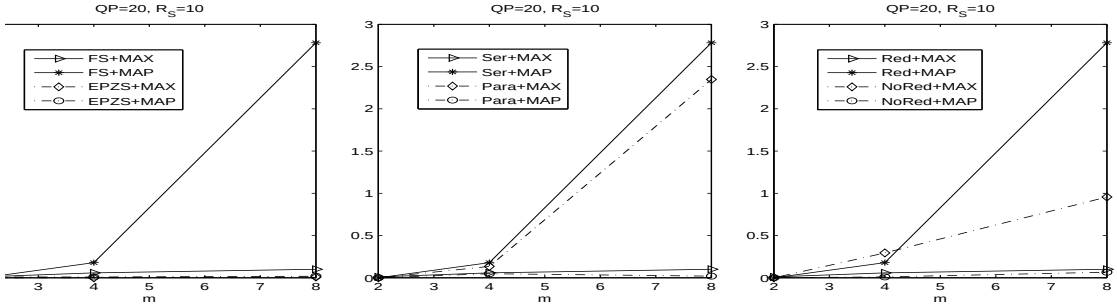

Fig. 3. Comparison of EPZS/FS ME algorithms, serial/parallel MMC architectures, redundant/non-redundant cases for $m = 2, 4, 8$, $R_S = 10$, and $QP = 30$

more balanced partial SAD dynamic range.

Note that there is a redundancy between VOS block and SS block; pixel input data to the SS block is a subset of input of VOS block. Thus, removing this redundancy, leads to lower power consumption as the number of operations in the VOS module (See Figure 1 (b)). Performance also increases because a smaller number inputs leads to lower probability of error in the VOS module.

5. REFERENCES

[1] M. A. Breuer, S. K. Gupta, and T. M. Mak. Defect and error tolerance in the presence of massive numbers of defects. *IEEE Design & Test of Comp.*, 21:216–227, May–June 2004.

[2] H. Cheong, I. Chong, and A. Ortega. Computation error tolerance in motion estimation algorithms. In *IEEE International Conference on Image Processing, ICIP'06*, Oct. 2006.

[3] H.-Y. Cheong and A. M. Tourapis. Fast motion estimation within the h.264 codec. In *Proc.IEEE Int. Conf. on MultiMedia and Expo (ICME-2003)*, July 2003.

[4] T. Chiang and Y. Q. Zhang. A new rate control scheme using quadratic rate distortion model. *IEEE Trans. Circuits Syst. Video Technol.*, 7(1):246–250, Feb. 1997.

[5] I. Chong and A. Ortega. Hardware testing for error tolerant multimedia compression based on linear transforms. In *Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, DFT'05*, pages 523–534, 2005.

[6] I. Chong and A. Ortega. Dynamic voltage scaling algorithms for power constrained motion estimation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr. 2007.

[7] H. Chung and A. Ortega. Analysis and testing for error tolerant motion estimation. In *Proc. of IEEE Int. Symp. on Defect Fault Tolerance in VLSI Syst.*, pages 514–522, 2005.

[8] E. Debes. Recent changes and future trends in general purpose processor architectures to support image and video applications. *Proceedings of the 2003 IEEE International Conference on Image Processing (ICIP'03)*, 3:85–88, 2003.

[9] F. Dufaux and F. Moscheni. Motion estimation techniques for digital tv: A review and a new contribution. *Proc. of the IEEE*, (6):858–876, June 1995.

[10] M. A. Elgamel, A. M. Shams, and M. A. Bayoumi. A comparative analysis for low power motion estimation VLSI architectures. In *IEEE Workshop on Signal Processing*, 2000.

[11] C. J. Hughes, J. Srinivasan, and S. V. Adve. Saving energy with architectural and frequency adaptations for multimedia applications. *34th Annual International Symposium on Microarchitecture (MICRO-34)*, 2001.

[12] P. Kuhn. *Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation*. Kluwer Academic Publishers, Boston, 1999.

[13] K. Lengwehasatit and A. Ortega. Probabilistic partial-distance fast matching algorithms for motion estimation. *IEEE Trans. Circuits Syst. Video Technol.*, 11(2):139–152, Feb. 2001.

[14] R. E. Lyions and W. Vanderkulk. The use of triple modular redundancy to improve computer reliability. Technical report, IBM J. Res. Develop., 1962.

[15] P. Pirsch, N. Demassieux, and W. Gehrke. VLSI architectures for video compression-a survey. In *Proc. IEEE*, volume 83(2), pages 220–246, Feb. 1995.

[16] K. L. Shepard and V. Narayanan. Noise in deep submicron digital design. In *ICCAD*, pages 524–531, Nov. 1996.

[17] P. Tseng, Y. Chang, Y. Huang, H. Fang, C. Huang, and L. Chen. Advances in hardware architectures for image and video coding- a survey. *Proc. of the IEEE*, (1):184–197, Jan. 2005.

[18] G. V. Varatkar and N. R. Shanbhag. Energy-efficient motion estimation using error-tolerance. In *International Symposium on Low Power Electronics and Design (ISLPED)*, 2006.